

Användare och utvecklare

-

om
anveckling
med
kalkylprogram

av

Anders Avdic

NÓ JOWNE (RE- SAN)

Dwon ó far
faes jowne beighth lass
lau ó gwyhn
ddwennë fae dí lothcair
nó naienth waien
nó dieyhn waien...



Natt och dag
min resa börjar här
ljus och mörker
jag skall nå fram
bortom dimman
bortom horisonten...

(John Avdic)

Innehållsförteckning

DEL I - GRUNDERNA

1	INTRODUKTION	3
1.1	Användare och utvecklare	4
1.1.1	Historik och utveckling	4
1.1.2	Formalisering	6
1.1.3	IT som möjliggörare	8
1.1.4	Sammanfattning av utvecklingstendenser	8
1.1.5	Kunskapsöverföringsproblemet	9
1.1.6	Kalkylprogram och kalkylsystem	12
1.2	Kalkylprogramanveckling	13
1.2.1	Relation mellan användning och utveckling	14
1.2.2	Anvecklare	15
1.2.3	Kunskap om utvecklingsverktyg	17
1.2.4	Verksamhetskunskap	18
1.2.5	Verksamhetsmotverkande chauvinism	19
1.2.6	Yrkesetik	21
1.2.7	Verksamhetsetik	22
1.2.8	Yrkes- och verksamhetsetik	22
1.2.9	Uttrycket Anveckling	23
1.3	Forskningsfråga	24
1.3.1	Preliminära förutsättningar för kalkylprogramanveckling	25
1.3.2	Tänkbara effekter	26
1.3.3	Formulering av forskningsfrågor	27
1.4	Syfte	28
1.5	Disposition	29
2	FORSKNINGSMETOD	31
2.1	Projekt och delprojekt	31
2.2	Strategi	32
2.3	En kvalitativ metodansats	35
2.3.1	Kvantitativ och kvalitativ	35
2.3.2	Metodtriangulering	37
2.4	En hermeneutisk metodansats	38
2.4.1	Förförståelse	39
2.4.2	Perspektivväxling	41
2.4.3	Intentionell aspekt	43
2.4.4	Subjektiv aspekt	45
2.5	En abduktiv metodansats	45
2.6	Empiri datainsamling	48
2.6.1	Teoretisk känslighet	49
2.6.2	Användning av ordbehandlare	51
2.6.3	Huvudvariabel	52
2.6.4	Relation till respondent, närhet eller distans?	52

2.6.5	Datainsamling och validitet	54
DEL II - EMPIRI		
3	INDUSTRI	57
3.1	Företaget	57
3.2	Genomförande	58
3.2.1	Metod	59
3.2.2	Urval av respondenter	59
3.2.3	Disposition av presentation	60
3.3	Ismo - produktionsplanerare	60
3.3.1	Uppgifter och verksamhetskunskap	60
3.3.2	Mål	60
3.3.3	Anveckling, frekvens och typ	61
3.3.4	Anveckling, exempel och analys	62
3.3.5	Arbetsätt	68
3.3.6	Sammanfattning - Ismo	70
3.4	Bosse - controller	71
3.4.1	Uppgifter och verksamhetskunskap	71
3.4.2	Mål	73
3.4.3	Anveckling, frekvens och typ	74
3.4.4	Anveckling, exempel och analys	74
3.4.5	Arbetsätt	80
3.4.6	Sammanfattning - Bosse	80
3.5	Thomas - avdelningschef	81
3.5.1	Uppgifter och verksamhetskunskap	81
3.5.2	Mål	81
3.5.3	Användning av kalkylprogram och presentationsprogram	82
3.5.4	Anveckling, exempel och analys	83
3.5.5	Sammanfattning - Thomas	84
3.6	Sammanfattning - Industri	84
4	MYNDIGHET	87
4.1	Myndigheten	87
4.2	Genomförande	88
4.2.1	Metod	88
4.2.2	Urval av respondenter	88
4.2.3	Disposition av presentation	89
4.3	Olle - exploateringsingenjör	89
4.3.1	Uppgifter och verksamhetskunskap	89
4.3.2	Mål	90
4.3.3	Anveckling, frekvens och typ	90
4.3.4	Anveckling, exempel och analys	91
4.3.5	Arbetsätt	98
4.3.6	Sammanfattning - Olle	99
4.4	Jan - kamrer och stabschef	100
4.4.1	Uppgifter och verksamhetskunskap	100
4.4.2	Mål	100

Innehållsförteckning

4.4.3	Anveckling, frekvens och typ	101
4.4.4	Anveckling, exempel och analys	101
4.4.5	Arbetsätt	105
4.4.6	Sammanfattning - Jan	106
4.5	Annette - administratör	107
4.5.1	Uppgifter och verksamhetskunskap	107
4.5.2	Mål	107
4.5.3	Anveckling, frekvens och typ	107
4.5.4	Anveckling, exempel och analys	108
4.5.5	Arbetsätt	110
4.5.6	Sammanfattning - Annette	110
	Sammanfattning - Myndighet	110
5	IT-SPECIALISTER	113
5.1	Genomförande	113
5.1.1	Metod	113
5.1.2	Urval av respondenter	114
5.1.3	Disposition av presentation	114
5.2	Hans - IT-chef	114
5.2.1	IT-miljö	115
5.2.2	Åtkomst till centrala data	115
5.2.3	Användningsområden för kalkylprogram	116
5.2.4	Problem med kalkylprogramanveckling	117
5.2.5	IT-avdelningen och verksamhetskunskap	118
5.3	Sten - systemutvecklare	118
5.3.1	IT-miljö	118
5.3.2	Åtkomst till centrala data	119
5.3.3	Användningsområden	119
5.3.4	Problem med kalkylprogramanveckling	120
5.3.5	Framtiden?	121
5.4	Mika - konsult	121
5.4.1	Problem med kalkylprogramanveckling	121
5.4.2	Problemorsaker och lösningar	123
5.5	Sammanfattning - IT-specialister	124
6	APPLIKATION	127
6.1	Genomförande	127
6.1.1	Metod	127
6.1.2	Miljö	128
6.2	Resultat	130
6.2.1	Systemfunktioner	130
6.2.2	Användningsfrekvens	130
6.2.3	Orsaker och effekter	131
6.2.4	Hantering	132
6.2.5	Beställares deltagande i utvecklingsarbetet	132
6.3	Diskussion	132
6.3.1	Användning	133
6.3.2	Utveckling	134

DEL III - REFERENS RAM

7	REFERENS MODELL	139
7.1	Strukturering av referensram	139
7.2	Kriterier för och kategorier i referensramen	140
7.3	Den praktikgeneriska modellen	141
7.4	Praktikgenerisk modell för utvecklare	145
7.5	Om referensramen	153
8	INFORMATIONSSYSTEM	155
8.1	Begreppet informationssystem	156
8.2	Traditionella informationssystem (TIS) och utvecklade informationssystem (a-system)	162
8.3	Kalkylsystem	163
8.4	Human-Scale Information System	165
8.5	Sammanfattning informationssystem kalkylsystem	170
9	SYSTEMUTVECKLING	171
9.1	Perspektiv på systemutveckling	171
9.2	Traditionell systemutveckling (TSU)	174
9.3	Livscykelmodellen	176
9.4	Livscykelmodellen - kalkylprogramutveckling	180
9.5	Sammanfattning Livscykelmodellen - kalkylprogramutveckling	183
10	ANVECKLING	185
10.1	Anveckling: definitioner och terminologi	186
10.2	Kalkylprogramutveckling	188
10.3	Förutsättningar för anveckling	191
10.3.1	Spridning av anveckling	191
10.3.2	Möjligt, berättigat, lämpligt	193
10.3.3	Möjlighet	193
10.3.4	Berättigande	194
10.3.5	Lämplighet	196
10.4	Effekter av anveckling	198
10.4.1	Systemoptimalitet - verksamhetskunskaper	198
10.4.2	Systemkvalitet - systemutvecklingsarbete	199
10.4.3	Informationskvalitet	200
10.4.4	Resursanvändning	201

Innehållsförteckning

10.4.5	Beslutsfattande och arbetssituation	202
10.4.6	Inflytande och styrning	203
10.4.7	Summering effekter - styrning eller ej	204
10.5	Arbetsätt	205
10.5.1	Traditionalism, organisation eller förnyelse?	206
10.5.2	Traditionalismsynsätt	207
10.5.3	Organisationssynsätt	209
10.5.4	Förnyelsesynsätt	210
10.6	Sammanfattning - Anveckling	211
11	AKTÖRER	213
11.1	Aktörer, traditionell systemutveckling	214
11.2	Anvecklaren	216
11.3	Aktörer, KPA-miljö	217
11.4	Anvecklare - en definition	220
12	INTENTION OCH BESLUTSSTÖD	221
12.1	Begreppet beslutsstödssystem	224
12.2	IT-utveckling ur ett beslutsstödsperspektiv	226
12.3	Arbetsuppgifter för beslutsstöd	229
12.4	Kalkylsystem - DSS	230
12.5	Kalkylprogramaspekter	230
12.6	Kommunikation - GDSS	231
12.7	Sammanfattning - Intention och beslutsstöd	232
13	KUNSKAP	235
13.1	Kunskapsformer	236
13.2	Verksamhetskunskap	237
13.3	Verktygskunskap för anvecklare	237
13.4	Kunskap, information och data	240
13.5	Svårformaliserbar kunskap	241
13.6	Tyst kunskap	243
13.6.1	En inledande innebördsdiskussion	244
13.6.2	Invändningar mot begreppet tyst kunskap	244
13.6.3	Olika aspekter av tyst kunskap	245
13.6.4	Fokal och tyst dimension	246
13.6.5	Objektivering av det tysta	247

Användare och utvecklare

13.7	Praktisk kunskap	248
13.8	Skill, know-how och kompetens	249
13.9	Professioner	250
13.10	Praktik, kunskap och handling	251
13.11	Alternativa ansatser	253
13.12	Sammanfattning - Kunskap och anveckling	254
14	NORMER	257
14.1	Normer och mål	259
14.2	Normer i relation till skill, know-how och kompetens	260
14.3	Mål- och värdebaserad praktik	262
14.4	Normer, kunskap och yrkesetik i professioner	264
14.5	Förändring av normer	267
14.6	Sammanfattning - Normer	267
15	UTVECKLINGSVERKTYG	269
15.1	Verktysbegreppet	270
15.2	Kalkylprogram	272
15.3	Kalkylprogram och risker	276
15.4	Användbarhet	279
15.5	Utvecklingsverktygs användbarhet	283
15.6	Cyklisk utveckling av verktyg	286
15.7	Sammanfattning - Utvecklingsverktyg	289
DEL IV - AVSLUTNING		
16	ANALYS AV ANVECKLINGSBEGREPPET	293
16.1	Kalkylsystem	293
16.2	Anveckling	294
16.2.1	Självförsörjning, gruppanveckling och fadderanveckling	295
16.2.2	Professionalism och kontroll	295
16.2.3	Integration på flera plan	295
16.2.4	Incitament till anveckling	296
16.2.5	Målstyrd anveckling	297
16.2.6	Dynamik och förändring	298

Innehållsförteckning

16.3	Intention	299
16.3.1	Olika syften	300
16.3.2	Förutsättningar för beslutsstöd	301
16.4	Underlag	302
16.5	Kunskap	303
16.5.1	Verksamhets-, professions- och verktygskunskap	303
16.5.2	Fördelning av kunskap	305
16.5.3	Objektivering av tyst kunskap	307
16.5.4	En annan typ av formalisering	308
17	VERKTYGSKUNSKAP OCH NORMER	313
17.1	Behov och möjligheter	314
17.1.1	Aktualitet	318
17.1.2	Noggrannhet	318
17.1.3	Formalisering och analys	319
17.1.4	Kontroll	320
17.1.5	Kommunikation	321
17.2	Det goda kalkylsystemet	321
17.2.1	Ett strukturerat system	322
17.2.2	Ett säkert system	322
17.2.3	Ett dokumenterat system	323
17.2.4	Ett flexibelt system	323
17.2.5	Ett komplexitetshanterande system	323
17.2.6	Ett system med utdata med god läsbarhet	324
17.2.7	Ett system med tillgång till data	325
17.2.8	Allmänna lösningar på problem	325
17.3	Sammanfattning – Utvecklingsverktyg och normer	326
18	SAMMANFATTANDE SLUTSATSER	327
18.1	Svar på forskningsfrågorna	327
18.2	Den anvecklarseriska modellen	329
18.3	Kontinuerlig förändring	330
18.3.1	Integration	331
18.3.2	Interaktivitet	331
18.3.3	Ifrågasättande	332
18.4	Metoder och förändring som permanent tillstånd	332
18.5	Framtida forskning	333
19	SUMMARY IN ENGLISH	335
19.1	Introduction	335
19.2	Methods	336
19.3	Empirical studies	336
19.4	Framework	337

Användare och utvecklare

19.5	Conclusions	339
	REFERENSER	341
	INDEX	353
	BILAGOR	359

Figurförteckning

FIGUR 1 FORMALISERBAR KUNSKAP MÅSTE FÖRST GÖRAS TILLGÄNGLIG OCH KOMMUNICERBAR.	6
FIGUR 2 UTVECKLINGEN VAD GÄLLER SYSTEMUTVECKLING GÅR MOT ATT MER OCH MER SVÅRFORMALISERADE UPPGIFTER KAN AUTOMATISERAS DÅ PROGRAMVAROR BLIR MER FLEXIBLA.	7
FIGUR 3 UTVECKLING OCH FÖRHÅLLANDE MELLAN FORMALISERINGSGRAD OCH TYP AV SYSTEMUTVECKLINGSMOMENT.	7
FIGUR 4 SCHEMATISK BILD AV KUNSKAPSÖVERFÖRING VID TRADITIONELL SYSTEMUTVECKLING (AVDIC 1995A:6).	10
FIGUR 5 KUNSKAPSÖVERFÖRINGSPROBLEMET VID TRADITIONELL SYSTEMUTVECKLING.	11
FIGUR 6 SCHEMATISK BESKRIVNING AV RELATION MELLAN ANVÄNDNING OCH UTVECKLING AV INFORMATIONSSYSTEM HOS DATORANVÄNDARKATEGORIER.	14
FIGUR 7 SCHEMATISK BESKRIVNING AV FÖRHÅLLET MELLAN VERKSAMHETSKUNSKAP OCH KUNSKAP OM UTVECKLINGSVERKTYG.	16
FIGUR 8 SCHEMATISK BESKRIVNING AV FÖRHÅLLET MELLAN KUNSKAP OCH UTVECKLING.	19
FIGUR 9 KALKYLPROGRAMANVECKLING: PRELIMINÄRA FÖRUTSÄTTNINGAR OCH INNEBÖRD.	25
FIGUR 10 SCHEMATISK BESKRIVNING AV AVHANDLINGENS ÖVERGRIPANDE SYFTE.	28
FIGUR 11 SAMB	30
FIGUR 12 ÖVERSIKTLIG ARBETSGÅNG FÖR FORSKNINGSPROJEKTET KALKYLPROGRAMANVECKLING.	31
FIGUR 13 PROCESS OCH PRODUKT: STATISK OCH DYNAMISK DIMENSION AV SAMMA FENOMEN.	42
FIGUR 14 OLIKA FOKUS PÅ VERKSAMHETSBEHOV OCH UTVECKLINGSVERKTYG.	43
FIGUR 15 DEDUKTION, INDUKTION OCH ABDUKTION (ALVESSON & SKÖLDBERG 1994:45).	46
FIGUR 16 SCHEMATISK BESKRIVNING AV KUNSKAPSUTVECKLINGSSTRATEGI.	47
FIGUR 17 SAMMANHANG OCH HELHET PÅVERKAR TOLKNING AV OBSERVATIONER (PATEL & TEBELIUS 1987:33).	47
FIGUR 18 FÖRFÖRSTÅELSE OCH ÖPPENHET.	49
FIGUR 19 MÖNSTER FÖR BEGREPPSMODELLER.	51
FIGUR 20 FÖRHÅLLET MELLAN INRE OCH YTTRE VALIDITET (SVENNING 1996:62).	54
FIGUR 21 SCHEMATISK BESKRIVNING AV PRODUKTIONEN PÅ DET STUDERADE FÖRETAGET.	57
FIGUR 22 SCHEMATISK BESKRIVNING AV DE ADMINISTRATIVA SYSTEMEN PÅ DET STUDERADE FÖRETAGET. (STRECKAD LINJE = EV ÖVERFÖRING)	58
FIGUR 23 KRAV PÅ SÄKRARE BESLUTSUNDERLAG GER KRAV PÅ NOGGRANNHET GER KRAV PÅ VERKTYG SOM STÖDJER DETTA.	63
FIGUR 24 EN ALLT MER FÖRÄNDERLIG OMVÄRLD STÄLLER KRAV PÅ VERKSAMHETSKUNSKAP I SYSTEMUTVECKLINGARBEDET.	63
FIGUR 25 KALKYLSYSTEM FÖR ALLOKERING AV ÖVERBLIVNA PALLAR.	64
FIGUR 26 EN KOMPLEX OCH KAPITALINTENSIV VERKSAMHET LEDER TILL BEHOV AV VERKSAMHETSKUNSKAP VID KALKYLPROGRAMANVECKLING.	65
FIGUR 27 DEL AV BLAD FÖR VISS EXTERNARKNARE I KALKYLSYSTEM FÖR UPPFÖLJNING AV EXTERNARKNING.	66
FIGUR 28 SUMMERINGSBLAD I KALKYLSYSTEM FÖR UPPFÖLJNING AV EXTERNARKNING.	66
FIGUR 29 KALKYLSYSTEM FÖR BERÄKNING AV ARKANTAL PER PALL.	67
FIGUR 30 KALKYLSYSTEM FÖR SIMULERING AV ARKNINGSFÖRDELNING.	67
FIGUR 31 KALKYLSYSTEM FÖR TARABERÄKNING.	69
FIGUR 32 DEL AV KALKYLSYSTEM FÖR JÄMFÖRELSE AV ARKNINGSKOSTNADER (SIFFRORNA ÄR FIKTIVA).	70
FIGUR 33 SCHEMATISK BESKRIVNING AV BOSSES INFORMATIONSHANTERING.	72
FIGUR 34 FÖRDELNING AV ARBETSUPPGIFTER PÅ EKONOMIAVDELNINGEN.	72
FIGUR 35 EKONOMIAVDELNINGEN OCH KALKYLSYSTEM I SAMVERKAN.	75
FIGUR 36 EFFEKTER AV LOKALT NÄTVERK OCH KALKYLPROGRAMANVECKLING.	76
FIGUR 37 FÖRUTSÄTTNINGAR FÖR RIMLIGHETSBEDÖMNINGAR.	76
FIGUR 38 DEL AV PRODUKTKALKYL (OBS FIKTIVA VÄRDEN).	77
FIGUR 39 FÖRENKLAD BESKRIVNING AV DEL AV KALKYLSYSTEM FÖR BERÄKNING AV TÄCKNINGSBIDRAG/TIMME.	78
FIGUR 40 FÖRUTSÄTTNINGAR FÖR FORMALISERING.	78

Användare och utvecklare

FIGUR 41 DEN KONTINUERLIGA FÖRÄNDRINGSCIRKELN. EFFEKTER AV KONTINUERLIGA FÖRÄNDRINGAR.	79
FIGUR 42 MÅL EFTERBEARBETNINGEN.	82
FIGUR 43 DEL AV KALKYLSYSTEM FÖR UPPFÖLJNING AV PRODUKTIONSDATA DYGNSSVIS.	82
FIGUR 44 DEL AV KALKYLSYSTEM FÖR PRESENTATION AV CENTRALA VARIABLER (OBS, FIKTIVA DATA).	83
FIGUR 45 FAKTAINSAMLING OCH UPPFÖLJNING AV VERKSAMHETEN.	84
FIGUR 46 DEN KONTINUERLIGA FÖRÄNDRINGSCIRKELN.	85
FIGUR 47 STRUKTUR PÅ EXPLOATERINGSSAMMANSTÄLLNING.	91
FIGUR 48 SYSTEM FÖR TIDRAPPORTERING.	92
FIGUR 49 DEL AV SYSTEM FÖR HYRESBERÄKNING VID INVESTERING.	93
FIGUR 50 DEL AV SYSTEM FÖR CASH-FLOW BERÄKNING.	94
FIGUR 51 DEL AV SYSTEM FÖR VÄRDERING AV TOMTPRISER.	95
FIGUR 52 DEL AV SYSTEM FÖR JÄMFÖRELSE AV PRIS- OCH INDEXUTVECKLING.	96
FIGUR 53 EFFEKTER AV MÖJLIGHET TILL NOGGRANT BESLUTSUNDERLAG.	99
FIGUR 54 EFFEKTER AV ÖVERSKÅDLIGHET OCH BEARBETNINGSBAR INFORMATION.	100
FIGUR 55 FÖRUTSÄTTNINGAR FÖR OCH EFFEKTER AV KALKYLPROGRAMANVECKLING PÅ MYNDIGHETEN.	102
FIGUR 56 RAPPORT OM INVESTERING PER VERKSAMHET.	103
FIGUR 57 DEL AV KALKYLSYSTEM FÖR BERÄKNING AV FASTIGHETSSKATT.	104
FIGUR 58 DEL AV KALKYLSYSTEM FÖR RAPPORTERING AV BRUTTOREDOVISNING MED ELLER UTAN INTERNTRANSAKTIONER.	105
FIGUR 59 EFFEKTER AV KALKYLPROGRAMANVECKLING.	107
FIGUR 60 MOTIV FÖR ANVÄNDNING AV KALKYLPROGRAM.	108
FIGUR 61 DEL AV SYSTEM FÖR PRESENTATION AV RESULTAT AV PERSONALENKÄT.	108
FIGUR 62 MOTIV FÖR VERKTYGSVAL FÖR SYSTEM FÖR PERSONALENKÄT.	109
FIGUR 63 DEL AV SYSTEM FÖR ANALYS AV BYGGLOV.	109
FIGUR 64 MOTIV FÖR VERKTYGSVAL FÖR SYSTEM FÖR BYGGLOVSSTATISTIK.	110
FIGUR 65 SCHEMATISK BESKRIVNING AV BUDGETERINGSSYSTEMET.	128
FIGUR 66 KOSTNADSSTÄLLEHIERARKIEN SCHEMATISKT.	129
FIGUR 67 EXEMPEL PÅ INMATNINGSBLAD I BUDGETSYSTEMET.	131
FIGUR 68 FÖRUTSÄTTNINGAR FÖR POSITIV UPPFATTNING AV BUDGETSYSTEMET.	134
FIGUR 69 EFFEKTER AV BUDGETSYSTEMET UR ANVÄNDARNAS PERSPEKTIV.	134
FIGUR 70 SLUTSATSER AV UNDERSÖKNINGEN.	135
FIGUR 71 DEN PRAKTIKGENERISKA MODELLEN (GOLDKUHL & RÖSTLINGER 1998:7).	142
FIGUR 72 RELATIONER MELLAN NORMER, KUNNANDE OCH VERKTYG (GOLDKUHL & RÖSTLINGER 1997:11).	144
FIGUR 73 ANVECKLARPRAKTIKEN INNEHÅLLER I SIN TUR MINST TVÅ PRAKTÍKER.	148
FIGUR 74 RELATIONER MELLAN NORMER, KUNNANDE OCH VERKTYG I EN ANVECKLARES PRAKTIK (MODIFIERING AV GOLDKUHL & RÖSTLINGER 1997:11).	151
FIGUR 75 FÖRDELNING AV KUNSKAP MELLAN PRODUCENT OCH VERKTYG.	152
FIGUR 76 MODIFIERING AV DEN PRAKTIKGENERISKA MODELLEN (MODIFIERING AV GOLDKUHL & RÖSTLINGER 1997:7).	153
FIGUR 77 DATASYSTEM I VERKSAMHET (GOLDKUHL 1993:16).	159
FIGUR 78 KALKYLSYSTEM ÄR EN TYP AV A-SYSTEM SOM ÄR EN TYP AV INFORMATIONSSYSTEM.	162
FIGUR 79 KALKYLSYSTEMTYPER OCH KALKYLPROGRAM (AVDIC 1995A:185).	164
FIGUR 80 OLIKA TYPER AV SYSTEMUTVECKLING (NILSSON 1991:57).	174
FIGUR 81 LIVSCYKELMODELLEN (ANDERSEN 1994:41).	177
FIGUR 82 KPA ÄR EN TYP AV ANVECKLING SOM ÄR EN TYP AV SYSTEMUTVECKLING.	186
FIGUR 83 OLIKA FOKUS PÅ ARBETSUPPGIFTER OCH SYSTEMUTVECKLING BEROENDE PÅ PERSPEKTIV (AVDIC 1995A:180).	189
FIGUR 84 OLIKA SYN PÅ ARBETSSÄTT FÖR ANVECKLING.	207
FIGUR 85 PRINCIPBESKRIVNING AV DATASYSTEM I VERKSAMHET (GOLDKUHL 1993:16).	216
FIGUR 86 FÖRHÅLLET MELLAN AKTÖRER I EN KPA-MILJÖ (AVDIC 1995A:49).	220
FIGUR 87 RELATIONER MELLAN SYFTEN MED KALKYLSYSTEM (AVDIC 1995A:191).	222
FIGUR 88 RELATION MELLAN TILLGÄNGLIG TID FÖR BESLUT OCH MÄNGDEN INFORMATION SOM KAN SAMLAS IN, KOMMUNICERAS OCH BEARBETAS.	224
FIGUR 89 RELATIONER MELLAN MIS, EIS, DSS OCH G(D)SS.	225
FIGUR 90 TRE NIVÅER AV DSS TEKNOLOGI (SPRAGUE & WATSON).	227

Figurförteckning

FIGUR 91 TILLGÅNG TILL IT LEDER TILL FÖRBÄTTRINGAR I KVALITATIV EFFEKTIVITET I KUNSKAPSUTVECKLING OCH BESLUTFATTANDE (HUBER 1990:70).	228
FIGUR 92 OLIKA TYPER AV KUNSKAP SOM EN ANVECKLARE BEHÖVER.	237
FIGUR 93 TAXONOMI FÖR ANVECKLARKUNSKAPER (PANKO 1988:167).	238
FIGUR 94 FÖRHÅLLANDE MELLAN KUNSKAP INFORMATION OCH DATA I ETT KOMMUNIKATIONSPERSPEKTIV.	240
FIGUR 95 OGDENS TRIANGEL (OGDEN & RICHARDS 1949)	241
FIGUR 96 FORMALISERINGSUPPGIFTER FÖR (1) VERKSAMHETSFÖRETRÄDARE OCH (2) UTVECKLARE UTIFRÅN OGDENS TRIANGEL.	242
FIGUR 97 VID TSU MÅSTE UNDERLAG FÖR SYSTEMUTFORMNING VARA KOMMUNICERBART.	242
FIGUR 98 VID ANVECKLING KAN FORMALISERINGSAKTIVITETER SKE UTAN ATT VISS INFORMATION, T.EX. UNDERLAG I FORM AV VERKSAMHETSKUNSKAP, GÖRS KOMMUNICERBAR.	243
FIGUR 99 TYST KUNSKAP SOM FUNDAMENT (ROLF 1991:67).	246
FIGUR 100 OBJEKTIVERING, ÖVERFÖRING AV KUNSKAP FRÅN DEN TYSTA TILL DEN FOKALA DIMENSIONEN.	247
FIGUR 101 I ANVECKLARENS PRAKTIK KAN HUVUDPRAKTIKEN GE UPPDRAG TILL UTVECKLARPRAKTIKEN.	253
FIGUR 102 OLIKA NORMER GÄLLER I ANVECKLARPRAKTIKENS OLIKA (DEL)PRAKTÍKER.	261
FIGUR 103 EXEMPEL PÅ PROFESSIONSNORMER I RELATION TILL HUVUDPRAKTIK OCH STANDARDISERING.	266
FIGUR 104 FUNKTIONER I EN INTERAKTIV KPA-MILJÖ.	275
FIGUR 105 OLIKA KUNSKAPSOMRÅDEN, RELEVANTA FÖR FÖREBYGGANDE AV FEL I KALKYLSYSTEM.	279
FIGUR 106 PRODUKTIVITET ENLIGT ALLWOOD (1990:11).	281
FIGUR 107 ANVÄNDBARHETSFAKTORER FÖR ANVECKLING.	286
FIGUR 108 BESKRIVNING AV DEN CYKLISKA UTVECKLINGEN AV VERKTYG	288
FIGUR 109 DELAR AV KALKYLPROGRAMMET INGÅR I KALKYLSYSTEMET.	294
FIGUR 110 ANVECKLING INNEBÄR INTEGRATION, INTERAKTIVITET OCH IFRÅGASÄTTANDE.	296
FIGUR 111 KONSEKVENSER AV ÖKAD VERKTYGSKUNSKAP.	296
FIGUR 112 TRADITIONELL SYN BEHOVSUPPFATTNING.	297
FIGUR 113 ANVECKLINGSSYN BEHOVSUPPFATTNING.	297
FIGUR 114 TRADITIONELL SYN PROBLEM- OCH MÅLUPPFATTNINGAR.	298
FIGUR 115 ANVECKLINGSSYN PROBLEM- OCH MÅLUPPFATTNINGAR.	298
FIGUR 116 OLIKA SYFTESTYPER MED KALKYLSYSTEM/KPA.	301
FIGUR 117 DELSYFTEN TILL BESLUTSSTÖD.	301
FIGUR 118 HÖGRE BESLUTSKVALITET MED ANVECKLING.	302
FIGUR 119 FORMALISERINGSGRAD FÖR OLIKA KUNSKAPSTYPER.	306
FIGUR 120 INDELNING AV KUNSKAP MED AVSEENDE PÅ TILLGÄNGLIGHET, KOMMUNICERBARHET OCH FORMALISERBARHET.	307
FIGUR 121 FÖRHÅLLET MELLAN ÖKAD VERKTYGSKUNSKAP OCH ÖKAD VERKSAMHETSKUNSKAP.	308
FIGUR 122 FÖRUTSÄTTNINGAR FÖR FORMALISERING GENOM INTERAKTIV UTVECKLING.	310
FIGUR 123 ANVECKLING I NÄTVERKSMILJÖ INNEBÄR MÖJLIGHET TILL VERIFIERING OCH IFRÅGASÄTTANDE.	311
FIGUR 124 ETT KALKYLSYSTEMEXEMPEL.	314
FIGUR 125 ANVECKLINGSAKTIVITETER.	317
FIGUR 126 INSPELADE MAKRON FÖR ATT TA BORT OCH SÄTTA DIT SKYDD.	319
FIGUR 127 KNAPPAR KOPPLADE TILL MAKRON I FIGUR 126.	320
FIGUR 128 EXEMPEL PÅ KONTROLL AV EN RÅVARAS ANDEL (FORMATERAD TILL PROCENT) AV TOTALSUMMA.	320
FIGUR 129 FÖRUTSÄTTNINGAR FÖR OCH EFFEKTER AV KPA.	329
FIGUR 130 DEN ANVECKLARGENERISKA MODELLEN.	330
FIGUR 131 DEN KONTINUERLIGA FÖRÄNDRINGSCIRKELN.	331
FIGUR 132 UTVECKLING OCH ANVECKLING.	333

Tabellförteckning

TABELL 1 IT:S INFLYTANDE PÅ PROCESSORIENTERAT FÖRÄNDRINGSARBETE (PROCESS INNOVATION) (DAVENPORT 1993:51).	8
TABELL 2 BEGREPP I ETT SAMMANHANG (ROHING 1978:39).	34
TABELL 3 KVANTITATIV OCH KVALITATIV METOD (HOLME & SOLVANG 1997:78).	35
TABELL 4 EMPIRISKA STUDIER I PROJEKTETS DEL TVÅ.	38
TABELL 5 AVHANDLINGENS INDELNINGSGRUNDER.	42
TABELL 6 KUNSKAPSBEHOV FÖR OLIKA TYPER AV ARBETSUPPGIFTER	124
TABELL 7 SKILLNADER MELLAN PROCEDURBASERAD OCH MÅLBASERAD PRAKTIK	146
TABELL 8 JÄMFÖRELSE AV HIS OCH KALKYLSYSTEM.	170
TABELL 9 EGENSKAPER FÖR SYSTEMUTVECKLING (ANDERSEN 1994:342).	172
TABELL 10 BESKRIVNING AV FASERNA I LIVSCYKELMODELLEN OCH DESS RESULTAT ENLIGT ANDERSEN (1994:43).	178
TABELL 11 KPA RELATERAT TILL LIVSCYKELMODELLEN (AVDIC 1995A:152F).	181
TABELL 12 JÄMFÖRELSE LIVSCYKELMODELLEN - KPA, EFTER AVDIC (1995A:156)	183
TABELL 13 INDELNING AV ANVECKLARE UTIFRÅN VEM SOM UTVECKLAR OCH ANVÄNDER A-SYSTEMET	217
TABELL 14 OLIKA TYPER AV UTVECKLARE (AVDIC 1995A:48).	219
TABELL 15 SKILLNADER MELLAN PROCEDUR- OCH MÅLBASERAT ARBETE (SPRAGUE & WATSON 1996:7).	229
TABELL 16 EFFEKTER PÅ KALKYLSYSTEM BEROENDE AV ANVECKLARENS KUNSKAPER OM KALKYLPROGRAM (EFTER WESTERLUND 1997:33).	240
TABELL 17 INDELNING AV PRAKTISK KUNSKAP (ROLF 1991:114).	249
TABELL 18 ABSTRAKTIONSGRAD FÖR IT-VERKTYG.	251
TABELL 19 MÅLKARAKTERISERING (GOLDKUHL & RÖSTLINGER 1988:86).	259
TABELL 20 ASPEKTER AV HANDLINGAR OCH DERAS INNEBÖRD (HABERMAS 1996:98).	262
TABELL 21 JÄMFÖRELSE: LANGEFORS DEFINITION AV IS OCH PANKOS DEFINITION AV KALKYLPROGRAM.	273
TABELL 22 KVANTITATIVA OCH KVALITATIVA FEL(MED UTGÅNGSPUNKT FRÅN PANKO & HALVERSSON (1996).	278
TABELL 23 MEKANISKA, LOKISKA OCH UTELÄMNANDE FEL.	278
TABELL 24 OSÄKRA INDATA OCH STRUKTURELLA FEL.	278
TABELL 25 OLIKA KONSEKVENSER AV ANALYS AV INFORMATION.	311
TABELL 26 KALKYLPROGRAMFUNKTIONER RELATERADE TILL OLIKA BEHOV.	317
TABELL 27 KALKYLPROGRAMFUNKTIONER RELATERADE TILL OLIKA AKTIVITETER.	318
TABELL 28 OLIKA TYPER AV KALKYLSYSTEM FÖR LOGISKA KONTROLLER.	320

Förord

När detta skrivs är fenomenet IT minst sagt aktuellt. Tidningar, radio, TV beskriver och analyserar varje dag förutsättningar och effekter vad gäller användning av datorer i olika sammanhang. Regeringar, kommuner och företag sammanträder, våndas och beslutar om olika åtgärder för att följa med i utvecklingen. Man tycker sig förstå, då man läser, lyssnar och tittar, att människor i västerlandet snart kommer att ha en näst intill obegränsad tillgång till information. En sak som inte så ofta berörs är den enskilda människans möjligheter att självständigt utnyttja, analysera och vidarebearbeta all information som hon så småningom kommer att ha tillgång till. Denna avhandling tar upp en aspekt av detta, nämligen anveckling, som handlar om människors möjlighet att tillgodose det informationsbehov, som inte direkt kan tillfredsställas genom surfande eller användande av standardprogram. Det tror jag kommer att bli en allt viktigare fråga under 2000-talet.

Att genomföra ett avhandlingsprojekt sker inte i en handvändning, åtminstone inte för mig. Åtskilliga är de vänliga själar som bidragit till att projektskeppet så småningom lägger till vid kaj efter en hel del kryssande och alltför mycket stiltje. Till mina kunniga, engagerade och trevliga handledare vill jag rikta ett stort tack. Först och främst, tack till professor Göran Goldkuhl, alltid kunnig, tålmodig, engagerad och idérisk. Göran har en formidabel förmåga att framföra kritik på ett konstruktivt och respektfullt sätt. Han kan engagera sig och samtidigt ställa krav och han har en förmåga att få en att tro att man kan klara av krävande uppgifter. Två egenskaper, som jag särskilt uppskattar är den respekt som Göran alltid visar och den avsaknad av fördömande, som utmärker hans handledning. Med min tunna akademiska bakgrund har denna attityd varit särskilt viktig. Görans fokusering på kvalitativ kunskapsutveckling och inget annat har varit den för mig största behållningen. Ett av de första råden som jag fick av Göran minns jag speciellt: "Kunskapsutveckling underlättas av prestigelöshet." Inte dumt. Tack också till professor Anders G Nilsson, som med sitt vänliga, kunniga och kompromisslösa sätt att ställa krav. Anders har alltid kommit med konkreta och konstruktiva förslag och på så sätt fört avhandlingsarbetet framåt, ibland då det verkligen behövts. Tack även till universitetslektor Ove L Johansson, som i slutskedet (ganska långt sådant) av avhandlingsarbetet tagit sig tid, läst flera versioner, kommenterat, diskuterat, föreslagit kloka förändringar samt delat med sig av sina värdefulla kunskaper och på så sätt gett ett stort och viktigt, för att inte säga avgörande, bidrag till denna avhandling.

Tack för värdefull granskning till (i bokstavsordning): Bengt E W Andersson, Riksrevisionsverket, Karin Axelsson, Linköpings universitet, Jenny Cederborg, Örebro universitet, Marie-Therese Christiansson, Karlstads universitet, Benneth Christiansson, Karlstads universitet, Owen Eriksson, Högskolan Dalarna, Klas Gäre, System Jönköping, Karin Hedström, Örebro universitet, Mikael Lind, Högskolan i Borås, Ulf Melin, Linköpings universitet, Lennart Ljung, Ericson Karlstad, Per Oscarsson, Örebro universitet, Markku Pellikka, Internationella handelshögskolan i Jönköping, Annie Röstlinger, Linköpings universitet, Pär Ågerfalk, Örebro universitet.

Användare och utvecklare

Tack för synpunkter och seminarieverksamhet till (i bokstavsordning): Jonas Arvidsson, Lars Nelander, Johan Petersson och Kenneth Åhlgren vid Örebro universitet.

Tack för insikter och medverkan till (i bokstavsordning): Anette Bjurström, Stadsbyggnadskontoret Örebro, Helen Borg, AssiDomän Frövi, Olle Emilsson, Stadsbyggnadskontoret Örebro, Per-Otto Emilsson, Nerikes Allehanda Örebro, Ingvar Helsing, AssiDomän Frövi, Hans Ingwald, Dyno Nobel Gyttop, Bo Magnusson, AssiDomän Frövi, Thomas Nyberg, AssiDomän Frövi, Tommy Petersson, AssiDomän Frövi, Ismo Sipola, AssiDomän Frövi, CG Svensson, Dyno Nobel Gyttop, Sten Tynander, AssiDomän Frövi, Mika Valjus, Martinsson Örebro, Jan Åkesson, Stadsbyggnadskontoret Örebro plus många fler, som under åren 1988 - 1999 bidragit direkt och indirekt med kunskaper och kalkylsystem.

Några alldeles särskilda tack går till:

- Gunnar Fors, som besparat mig mycket arbete med praktisk dokumenthantering när Microsoft inte klarade att lösa ordbehandlingsproblem.
- Tom Lacy, som kan rycka in när ens kunskaper i engelska inte räcker till.
- Fredric Westerlund, som tack vare sitt examensarbete om kalkylsystem, förorsakat många givande diskussioner och uppslag.

En tacksamhetens tanke flyger till min gamle prefekt Åke Holmén, som väl knappast trodde på någon himmel, men som helt säkert nu skulle suttit och myst i denna himmel, (om den funnits) då hans ambitioner att lägga en vetenskaplig grund för informatikämnet i Örebro, tagit ytterligare ett stapplande steg framåt.

Tack för inspiration och många kloka frågor och svar till kloka och trevliga arbetskamrater och studenter vid Örebro universitet. Näst familjen, är den sociala gemenskap, som arbetskamraterna utgör den viktigaste förutsättningen för ett harmoniskt liv, som är en förutsättning för att hålla långlivade projekt igång. Här är jag lyckligt lottad. Vem kan ha bättre arbetskamrater? En annan arbetsförmån som är få förunnad är den som man som universitetslärare har när man får umgås med vetgiriga studenter på arbetstid.

Tack för administrativ skicklighet och trevliga pratstunder till Lillemor Wallgren m.fl. vid Linköpings universitet samt Gurssi Nodklint Örebro universitet.

Tack till berörda beslutsfattare vid Högskolan i Örebro/Örebro universitet, som varit vänliga och tålmodiga nog att finansiera mina forskarstudier. Förhoppningsvis har investeringen gett och kommer att ge viss avkastning.

Tack till min stora familj för att de låtit mig hållas tills detta långa projekt äntligen avslutats. Om det inte har varit direkt kul att skriva avhandling, så har det i alla fall varit utvecklande och stimulerande på många andra sätt. Kanske kommer jag rent av att sakna mitt dåliga samvete och min oantastliga ursäkt för att slippa olika uppgifter, som t.ex. avfrostning av frysen, kavajinköp, gräsklippning, halltapetsering, buskflytt, läxförhör, hämtning av hönsmat etc. etc. Och kanske kommer jag nu också rent av att till-

Förord

låta mig att spela en patiens eller två, vilket kan komma att leda till roliga men ineffektiva stunder. Vetenskap i all ära, men vad vore livet värt utan lite ineffektiv irrationalitet och en god middag med sina nära och kära då och då?

Några begreppsanvändnings- och formaliaaspekter bör nämnas. I avhandlingen används pronomenet *han* vid syftning på anvecklare. Mer relevant vore att skriva han/hon, eftersom anveckling inte är knutet till kön. Detta skrivsätt föreföll mig lite otympligt varför jag valt att konsekvent välja han. Majoriteten av respondenterna är trots allt män.

I avhandlingen har referenser normalt försetts med sidreferenser. Inte minst för min egen skull. I de fall som sidhänvisning saknats kan det bero på att hela referensen och inte bara vissa sidor berörts. I dessa fall har sidhänvisning utelämnats. Någon gång beror det på att dokument hämtade från nätet inte har sidnumrering.

Alla exempel på kalkylsystem är utförda i MS Excel 97.

Avslutningsvis inledningsvis skall några väsentliga ståndpunkter understrykas (med flera streck). Anveckling kan fungera som ett komplement och i vissa fall ett alternativ till traditionell systemutveckling (TSU). Anveckling kan givetvis aldrig ersätta TSU. I de flesta fall av anveckling, är TSU aldrig aktuell. Denna avhandling syftar inte till att ifrågasätta existensen av TSU. Det riktas ingen ifrågasättande kritik mot TSU som sådan. Jag har själv bedrivit TSU och inser att denna verksamhet är nödvändig och kommer att bestå under överskådlig tid. Det finns i avhandlingen ingen ambition att framställa systemutvecklare som annat än viktiga, professionella yrkespersoner med minst lika goda avsikter, som andra professionella, t.ex. forskare.

Adolfsberg, september 1999
Anders Avdic

Förkortningar

A-system	Anvecklat system
AIS-K	Arbetsintegrerad systemutveckling med kalkylprogram
D1	Delprojekt 1, avslutat med licentiatavhandling
D2	Delprojekt 2, avslutat med denna doktorsavhandling
DSS	Decision Support System
EUC	End User Computing
GT	Grounded Theory
IS	Information System
KPA	Kalkylprogramanveckling
LAN	Local Area Network (Lokalt nätverk)
SP	Spreadsheet Program
SP-UDA	Spreadsheet Program - User Developed Application
TIS	Traditionellt informationssystem / Traditional Information System
TSD	Traditional Systems Development
TSU	Traditionell systemutveckling
UD	User Development
UDA	User Developed Application
USD	User Systems Development

Del I

DEL I

Grunderna

”Den övergripande forskningsfrågan för denna avhandling kan formuleras:
Vilka nya möjligheter får användare att utföra arbetsuppgifter då de själva kan bygga informationssystem?”
(Anders Avdic, 1999:27)

Syftet med detta kapitel är:

- *att redovisa bakgrunden till varför denna avhandling skrivs,*
- *att introducera för avhandlingen grundläggande och relevanta begrepp*
- *att presentera de bakomliggande forskningsfrågorna och syftet med avhandlingen*

1 Introduktion

Första gången jag träffade en anvecklare, var 1988. Anvecklaren var en budgetchef, som med kalkylprogram som utvecklingsverktyg, själv hade gjort ett budgeterings-system av icke ringa storlek. Han hade inga programmerings- eller systemeringskunskaper. Hans utvecklingsarbete möjliggjordes av verksamhetskunskap och viss kunskap om kalkylprogram. Ur min systemvetenskapliga synvinkel framstod detta som osannolikt. Kunde man verkligen bygga system utan systemerare och metoder? Sedan dess har jag träffat åtskilliga personer som, utan att reflektera över att det varit systemutveckling som de ägnat sig åt, byggt system av skiftande slag.

Användarnas roll i IT-världen¹ har förändrats. Från att ha varit en ganska diffust definerad grupp människor, förutbestämda att leva med informationssystem, som systemutvecklingsspecialister konstruerat, har användarna (via olika användarvänliga systemutvecklingsansatser) hamnat i en situation, som innebär att de själva kan utveckla egna informationssystem. Användarna har blivit utvecklare. Detta innebär nya intressanta förutsättningar för systemutveckling och därmed för verksamheter med informationssystem, dvs i stort sett alla verksamheter. Det är vad denna avhandling handlar om.

¹ Med IT avses i avhandlingen informationsteknik i en vid betydelse.

1.1 Användare och utvecklare

Användare och utvecklare innehar huvudrollen i såväl datorernas historia, som i denna avhandling. Detta avsnitt syftar till att lyfta fram huvuddragen i det förhållande, som funnits mellan dessa båda roller sedan systemutveckling började bedrivas i och med att den första datorn togs i användning på 1940-talet.

1.1.1 Historik och utveckling²

För att använda den första generella elektroniska datorn, ENIAC³ från 1946, krävdes personer med avancerad specialistkompetens. Programmering skedde genom uppkoppling av kablar på en panel. Byte av program kunde ta flera arbetsdagar. Förutom att ENIAC var svår att använda, var den även omfångsrik. ”Maskinen hade 18 000 elektronrör, var 26 meter lång, vägde 30 ton och hade en effektförbrukning på 140 kW” (Nationalencyklopedin). Användningsområdet för ENIAC var uträkning av projektilbanor för försvaret. Maskinen hade en prestanda, som var underlägsen en av dagens fickräknare. För att kunna använda ENIAC var man tvungen att förstå maskinens uppbyggnad, vilket kan jämföras med att man skulle vara tvungen att förstå förbränningsmotorerna och kraftöverföringens principer och konstruktion för att kunna köra bil. Det kan också jämföras med om bilkonstruktören var den enda som kunde använda bilen. De första datortillämpningarna syftade följaktligen inte till att tillverka program, som skulle användas av andra än de som gjort programmen (och datorn).

Genom olika tekniska framsteg och idéer och genom utvecklingen av datorer har det skapats nya förutsättningar för utveckling och användning av informationssystem i en takt som överstigit många bedömares prognoser. Månadstidningen *Popular Mechanics* skrev t.ex. 1949: ”*Computers in the future may weigh no more than 1,5 tons*”. Via mer och mer sofistikerade programspråk, via elektroniska kretsar präglade av *miniatyrisering, minskande energiåtgång, hastighetsökning och förbilligande*, via människors ökande kunskaper om datorer och deras användning, via utveckling av kringutrustning, via utvecklade datakommunikation mm har *möjligheten för icke-specialister att utnyttja datorn som verktyg ökat* för varje år som gått sedan 1946 års ENIAC.

En effekt av ovan beskrivna utveckling är att IT i form av datorer, annan utrustning, programvara och datakommunikation, under vissa omständigheter, kan integreras i verksamheter och inte enbart, såsom tidigare varit fallet, hanteras av specialister på datacentraler e.d. Det finns idag tack vare ovan beskrivna utveckling allt större möjligheter för verksamhetsföreträdare att med hjälp av datorer kontrollera den information som efterfrågas, behandlas eller produceras av företrädarens verksamhet. Det sker följaktligen en utveckling mot *integration av verksamhetsansvar och systemutvecklingskompetens*.

Under 1960- och 1970-talet har informationssystem i företag företrädesvis utvecklats och administrerats centralt. Skälet till detta är att det krävts specialkompetens för att

² Huvudreferens till avsnittet är Avdic 1995a. I övrigt anses fakta som allmänt kända.

³ Konstruerad av Eckert och Mauchly vid University of Pennsylvania.

Introduktion

handha utrustning och för att utveckla system. I och med utvecklingen mot prestandaökning, prissänkning, miniatyrisering och sänkning av energiåtgång sker en utveckling mot *fler lokala system*⁴.

En viktig grund för de lokala system som beskrivs ovan är introduktionen av persondatorn som ägde rum i början av 1980-talet. *Ökningen av antalet persondatorer* har lett till att många personer kommer i kontakt med datorer och olika typer av programvaror i större utsträckning än vad som tidigare varit fallet. Denna ökning av användningen har i sin tur lett till ökad utveckling av programvaror för persondatorer.

I och med att fler och fler använder datorer och utbudet av programvaror ökar, *ökar möjligheterna att utnyttja standardlösningar* i stället för att konstruera helt nya program.

På 1960- och 1970-talet då stor- och minidatorer behärskade marknaden hade varje datormärke ett eget operativsystem och följaktligen en egen uppsättning programvaror med begränsade möjligheter att kommunicera med andra miljöer med andra operativsystem. I takt med persondatorernas utbredning har större möjligheter att utbyta data över datormiljögränserna i nämnda mening erhållits. Man kan säga att det sker en *integration av datormiljöer*.

Inom respektive miljö har det även skett en *integration av programfunktioner*. Det är inte ovanligt att man i ett program kan hitta funktioner för såväl ordbehandling, grafik, kalkyl och makroinspelning. I något fall har detta lett till en sammansmältning av programvaror så att man som användare med en musklickning kan kalla på t.ex. grafikfunktioner i ett kalkylprogram⁵

En *standardisering av gränssnitt* mellan människa och dator har också ägt rum under senare år. Spridningen av så kallade grafiska gränssnitt har lett till att man som användare har lättare att känna igen sig i olika miljöer än tidigare, då olika leverantörer hade olika gränssnitt. Speciellt på persondatormarknaden har grafiska gränssnitt av Windows-karaktär fått stor spridning. Spridningen av datorer och datorprogram har underlättats av en utveckling mot användning av *språkformer* i gränssnitten mellan människa och dator *som är mer lättillgängliga* för människor som inte är IT-specialister. Detta gäller såväl användning som utveckling av informationssystem⁶. Då det gäller t.ex. kalkylprogram erbjuds möjligheter att skapa informationssystem på *deklarativ* och *direktmanipulerande* väg⁷. Att bygga ett kalkylsystem är ett exempel på detta.

Datakommunikation är ett område som fått ökad uppmärksamhet under senare år. Möjligheten att koppla samman datorer ökar tack vare uppbyggnad av datakommuni-

⁴ Med lokalt system menas ett informationssystem som stödjer en avgränsad verksamhet, t.ex. en logistikavdelning eller en inköpsavdelning.

⁵ Ett exempel på detta är OLE 2.0 som finns i MS Windows.

⁶ Jämför med benämningen på programspråksgenerationer 3GL, 4GL osv.

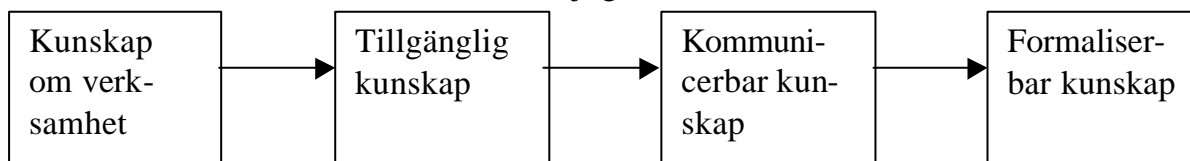
⁷ I motsats till procedurell teknik dvs programmering med procedurella språk av typen Cobol, C e.d.

kationsnät. Tillgängligheten till datorkraft och information kommer att öka i och med ökad förekomst av lokala och globala datakommunikationsnät.

På 90-talet använder barn datorer i skolan. Användande av datorer på arbetsplatser och i hemmen ökar. I nyhetsprogram och tidningar blir informationstekniken som fenomen regelbundet belyst. Dessa faktorer tillsammans innebär att *kunskap om datorer och informationsteknik ökar* bland människor.

1.1.2 Formalisering

Automatisering och *formalisering* är begrepp som är starkt relaterade till utvecklingen inom IT-området. "Att automatisera innebär att få någonting att gå av sig självt. Syftet kan vara att avlasta människan arbete och risker men också att höja effektiviteten i en process. Automatisering tillämpas på alla mänskliga aktivitetsområden." (Nationalencyklopedin) Med hjälp av datorprogram kan olika typer av aktiviteter automatiseras. För att automatisering skall vara möjlig krävs formalisering av aktiviteterna. Med formalisering avses en "...process varigenom formen hos något blir alltigenom specificerad" (Nationalencyklopedin). Formalisering kan ske av såväl dynamiska som statiska aspekter av verksamheter. Formalisering av dynamiska aspekter kan vara flödesdiagram, programkod e.d. Formalisering av statiska aspekter kan vara objektmodeller, postbeskrivningar e.d. Formalisering kan ske i olika grad. I IT-sammanhang är formaliseringar som kan datorbearbetas av särskilt intresse. Exempel på sådan formalisering är matematiska formler. En typ av formalisering är den som sker vid systemutveckling när systemutvecklare tillgodogör sig *tillgänglig kunskap* om verksamheter för att sedan formalisera en del av denna kunskap i form av informationssystem. Kunskap om verksamheter måste om inte systemutvecklaren har den själv, i dessa fall göras tillgänglig och kommunicerbar, för att kunna formaliseras (se Figur 1 nedan). Denna formaliseringsprocess kan vara olika komplicerad. Viss kunskap, t.ex. tyst kunskap (se kap 13.6), kan vara svår och ibland t.o.m. omöjlig att formalisera.



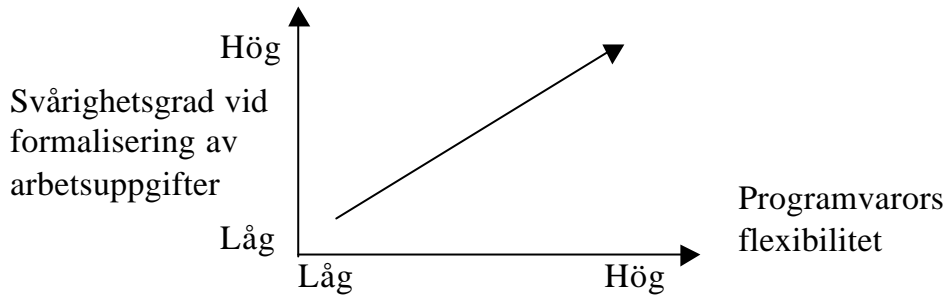
Figur 1 Formaliserbar kunskap måste först göras tillgänglig och kommunicerbar.

De första administrativa informationssystemen syftade till att automatisera vad som ibland kallas för smör- och brödrutiner, t.ex. lagersystem och lönesystem. Tack vare den utveckling som beskrivits i det inledande avsnittet, kan datorer idag utnyttjas på ett mindre (i förväg) formaliserat sätt. Kalkylprogram används t.ex. ofta för att skapa spontana uppföljningar eller beräkningar i avsikt att generera beslutsunderlag. Möjligheterna att *tillgodose spontana icke-rutinmässiga informationsbehov* kommer följaktligen att öka.

De ovan nämnda tidiga datortillämpningarna var den typ av administrativa rutiner som var enklast att automatisera i och med att de var formaliserbara. Att beräkna löner eller

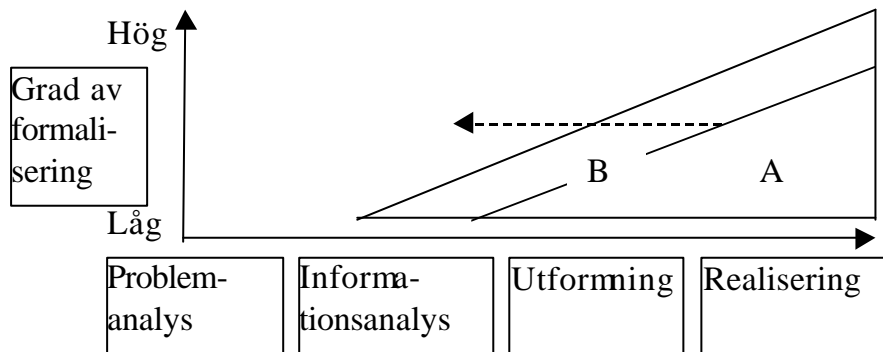
Introduktion

lagersaldon, kan beskrivas med regler. Svårare är att t.ex. hitta regler för beslut om vilka åtgärder som skall sättas in i produktionen för att kundens kvalitetsbehov skall bli bättre tillgodosett. Med programvaror, som kan användas flexibelt av personer med verksamhetskunskap *kan vissa mer svårformaliserade verksamhetsbehov tillgodoses* (se Figur 2 nedan).



Figur 2 Utvecklingen vad gäller systemutveckling går mot att mer och mer svårformaliserade uppgifter kan automatiseras då programvaror blir mer flexibla.

Programvarors utveckling tenderar att formalisera och stödja systemutvecklingsaktiviteter allt tidigare under systemutvecklingsprocessen⁸. I Figur 3 nedan visar den mindre triangeln (A) en tidigare situation där programvaror huvudsakligen stödde de konstruktionsorienterade momenten i systemutveckling (kompilatorer, interpretatorer etc.). Den utökade triangeln (B) visar hur programvaror allt mer stöder de mer analytiska momenten i systemutveckling (CASE-verktyg). Detta visar en tendens (visas med den streckade pilen) att *IT stödjer allt mer svårformaliserade aktiviteter under systemutvecklingsprocessen*.



Figur 3 Utveckling och förhållande mellan formaliseringsgrad och typ av systemutvecklingsmoment.

Avhandlingen utgår från att de utvecklingstendenser som beskrivs i ovanstående två avsnitt kommer att fortsätta. Innebörden av denna utveckling är inte lätt att förutse men medför möjligheter och problem som vi idag endast kan gissa. Det faktum att utvecklingen fortsätter och kanske accelererar är viktigt att ta hänsyn till såväl inom näringsliv och myndigheter som inom utbildning och forskning.

⁸ Om den betraktas ur ett livscykelperspektiv. Detta behandlas närmare i kapitel 9.

1.1.3 IT som möjliggörare

Beskrivningen ovan har delvis utgått från en traditionell syn på systemutveckling och IT, som ett sätt att tillgodose befintliga behov i en verksamhet. En annan syn, ser IT som *möjliggörare (enabler)* (Davenport 1993:16), vilket skulle innebära att fenomen inom IT, t.ex. kalkylprogram eller Internet i sig kan stimulera till förändringar och vara incitament till utvecklingsaktiviteter av olika slag. Dessa utvecklingsaktiviteter är då inte primärt resultatet av upplevda informationsbehov utan av upplevd möjlighetspotential hos olika IT-fenomen. Denna syn på IT innebär att IT-kompetens i form av t.ex. en dataavdelning inte enbart betraktas som en stabsfunktion på ett företag utan som något som kan generera centrala affärsidéer.

Davenport klassificerar betydelsen av IT:s inflytande på processororienterat förändringsarbete⁹ i nio kategorier (se Tabell 1 nedan). Davenports kategorisering syftar till att visa vilka olika typer av inverkan IT kan ha vid förändringsarbete. Då IT fungerar som *enabler* kan i en verksamhet åstadkommas effekter, som annars inte vore möjliga. I vissa branscher kan helt nya affärsområden skapas. Ett exempel på detta är bankbranschen, vars verksamhet förändrats till väsentliga delar i och med ökad användning av IT. Möjligheter för bankkunder att uträtta bankärenden utan att besöka banken är ett exempel på detta.

Tabell 1 IT:s inflytande på processororienterat förändringsarbete (*process innovation*) (Davenport 1993:51).

Inverkan	Förklaring
• Automatisering	Eliminera mänskligt arbete från processer
• Information	Fånga processinformation
• Sekvensering	Ändra processekvens, eller möjliggöra parallella processer
• Kontroll	Övervaka processers status och processobjekt
• Analytisk	Förbättra analys av information and beslutsfattande
• Geografisk	Koordinera geografiskt åtskilda processer
• Integrerande	Koordinera uppgifter and processer
• Intellektuell	Fånga och fördela intellektuella tillgångar
• Rationalisering	Eliminera onödiga delar av processer

1.1.4 Sammanfattning av utvecklingstendenser

Här följer en sammanfattning av ovan beskrivna utvecklingstendenser formulerade som antaganden om vilka förutsättningar som kommer att gälla för datoranvändning, vilket innefattar såväl utveckling som användning, under de närmaste åren:

- Den hittillsvarande utvecklingen av informationsteknik vad gäller prestandaökning, prissänkning, miniaturisering och sänkning av energiåtgång kommer att fortsätta.
- Det kommer att krävas allt mindre IT-specialistkompetens för att utveckla informationssystem.

⁹ "Process innovation".

Introduktion

- Det kommer att ske en utveckling, mot integration av verksamhetsansvar och systemutvecklingskompetens.
- Informationssystem kommer i framtiden allt oftare att utvecklas och administreras lokalt, sk lokala system.
- Antalet persondatorer kommer att öka.
- Möjligheterna att utnyttja standardlösningar kommer att öka.
- Det kommer att ske en fortsatt integration vad gäller datormiljöer.
- Det kommer att ske en fortsatt integration av programfunktioner.
- Det kommer att ske en utveckling mot mer lättillgängliga språkformer vid användning och utveckling av informationssystem.
- Det kommer att ske en utveckling mot icke-procedurell formalism vid utveckling av informationssystem.
- Tillgänglighet till datorkraft och information kommer att öka bl.a. genom fler lokala nät och ökad användning av globala nät.
- Utveckling av informationssystem kommer alltmer att handla om formalisering av icke-rutinmässiga arbetsuppgifter till skillnad från rutinmässiga uppgifter av typen löneberäkningar.
- Kunskap om informationsteknik kommer att öka bland människor i allmänhet.
- Nya tillämpningar inom informationstekniken kommer att se dagens ljus och vara att räkna med på sätt som idag inte kan förutses.
- En kvalitativt annorlunda aspekt är att IT kan fungera som möjliggörare för förändringsarbete i verksamheter och inte enbart som ett medel att tillgodose uppfattade behov.

1.1.5 Kunskapsöverföringsproblemet

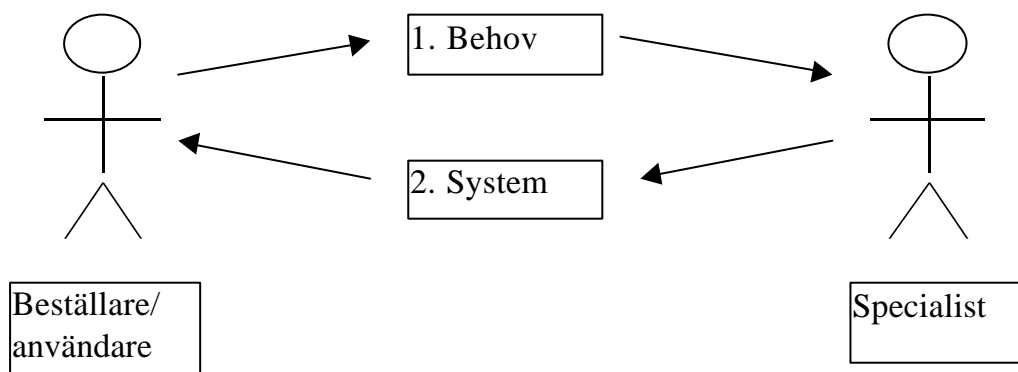
Utvecklingsarbete har traditionellt utförts av IT-specialister. Skälet har bl.a. varit att arbetet har krävt specialkunskaper som inte varit enkla att tillägna sig. Detta har vid systemutvecklingsarbete lett till en uppdelning av roller i *IT-specialister* och *verksamhetsföreträdare*¹⁰. Verksamhetsföreträdarna kallas ofta användare. De har, i en systemutvecklingskontext, haft till uppgift att förse utvecklarna med information om vilka behov som funnits. Utvecklarna har haft till uppgift att tolka denna information för att omvandla den till programkod. Enligt Langefors resonemang om den infologiska ekvationen (Langefors 1993:150f) är detta en icke-trivial uppgift. Langefors beskriver ekvationen $I = i(D, S, t)$; där I är den information (eller kunskap) som produceras från data D och förförståelse¹¹ S , genom tolkningsprocessen i , under tiden t . Det som Langefors kallar förförståelse (S) och som kan ses som en människas referensram, varierar från människa till människa beroende på erfarenheter, värderingar, kulturell bakgrund, fysiska och psykiska förutsättningar etc. Att till fullo förklara en mer eller mindre komplex situation för en människa så att denna förstår alla dimensioner på samma sätt som den som förklarar blir enligt detta resonemang mer eller mindre

¹⁰ Verksamhet ges här en vid betydelse och kan omfatta organisationer, företag, avdelningar eller mer avgränsade funktioner eller aktiviteter. Verksamhetsföreträdare är involverade i verksamheten och har kunskaper om denna. Verksamhetskunskaperna är av intresse i systemutvecklingssammanhang.

¹¹ "Preknowledge".

omöjligt enligt ekvationen. Vi har som människor skilda, fast likartade referensramar, olika representationsnormer och olika mål, vilket vid överföring av kunskap kan skapa problem i olika sammanhang t.ex. vid systemutveckling (Goldkuhl, Nilsson & Röstlinger 1982:8). Följden av dessa omständigheter är att problem kan uppstå vid överföring av kunskap mellan intressenter i ett systemutvecklingsprojekt. Detta resonemang syftar inte till att döma ut möjligheterna att kommunicera i systemutvecklingsprojekt. I den praktiska situationen står inte människor oförstående inför varandras försök att kommunicera. Ändamålsenliga informationssystem har byggts och kommer att byggas även om användare och utvecklare inte har en fullständig insikt i varandras referensram. Icke desto mindre, kan det faktum att kommunikation i ett systemutvecklingsprojekt kan vara problematisk, vara ett gott skäl till att fokusera problemet, vilket görs nedan.

Traditionell systemutveckling (TSU) (se även kap 9) innebär att då ett informationssystem skall byggas, engageras en eller flera personer från en specifik funktion, avdelning eller organisation, vars huvudsysselsättning är att utforma informationssystem. För dessa personer beskrivs informationsbehov och andra omständigheter så tydligt och heltäckande som möjligt. Specialisterna bygger sedan systemet iterativt och användaren tar det i bruk (se Figur 4 nedan).



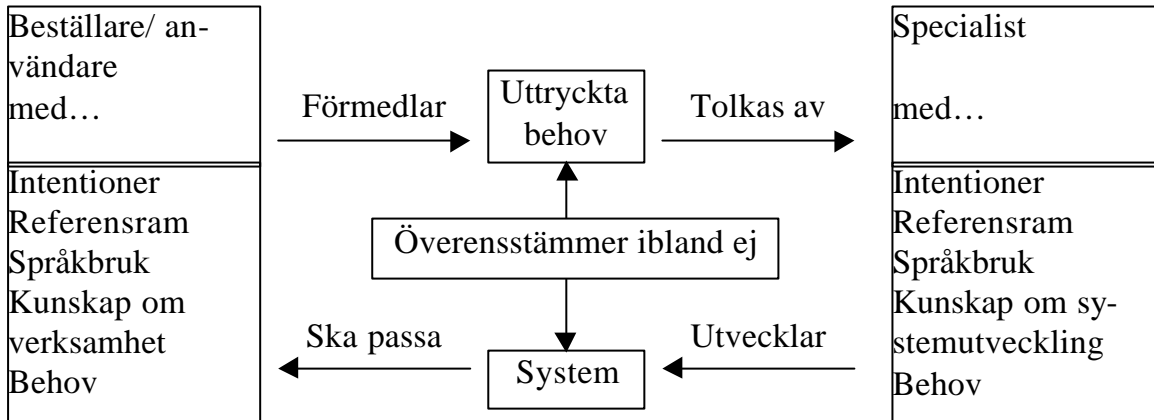
Figur 4 Schematisk bild av kunskapsöverföring vid traditionell systemutveckling (Avdic 1995a:6).

Ett centralt problem vid TSU är just att överföra kunskap om olika förhållanden mellan de olika intressenterna. *Kunskapsöverföringsproblemet* gör sig gällande i två riktningar, dels från beställare/användare till specialister dels från specialister till beställare/användare (se Figur 5 nedan). När det gäller den första riktningen (1. Behov) gäller det att beställarens/användarens uppfattningar om informationsbehov utifrån verksamhetens karaktär och mål kan vara svåra att förmedla till andra personer utan ingående kännedom om den aktuella verksamheten. Specialisterna kan ha andra intentioner, annan referensram, annat språkbruk och annan förståelse för och kunskap om verksamheten (Goldkuhl, Nilsson & Röstlinger 1982:9). Å ena sidan kan medverkan av professionella systemutvecklare ge en katalytisk positiv effekt på användares behovsförståelse och förmåga att utveckla och uttrycka behov. Å andra sidan kan olika omständigheter innebära att behoven kan vara svåra att uttrycka på ett ändamålsenligt sätt eller att det aldrig ges tillfälle att uttrycka dessa behov. Sådana omständigheter kan vara att behov inte bedöms som tillräckligt omfattande för att leda till initiering av ett

Introduktion

traditionellt systemutvecklingsprojekt, att behov förändras, att projekt avgränsas eller att ekonomiska förhållanden påverkar projektriktning eller projektplaner. Ytterligare en omständighet kan vara att verksamhetsföreträdaren kan ha svårt att formulera sina behov på ett sätt som systemutvecklaren kan tolka eller förstå.

När det gäller den andra riktningen (2. System) så gäller att specialisterna, efter att ha tolkat beställarens/användarens behov, skall formalisera detta i form av ett informationssystem, vilket sedan implementeras hos beställaren/användaren. Specialistens problem är att utforma ett system som passar användarens intentioner, referensram, språkbruk och behov.



Figur 5 Kunskapsöverföringsproblemet vid traditionell systemutveckling.

I denna schematiskt beskrivna situation kan två huvudtyper av kunskap identifieras. Å ena sidan *verksamhetskunskap* och å andra sidan *kunskap om utveckling av informationssystem*. Vid TSU är framför allt den förra sorten av kunskap föremål för överföring. Visserligen kan användare mer eller mindre systematiskt tillägna sig kunskaper om systemutvecklingsmetoder, databasdesign, programmering o.d. Men ett av de starkaste skälen till att det finns specialister är just att specialistkunskaper är komplicerade och/eller tidsödande att tillägna sig. Det är följaktligen specialisten som har att tillägna sig verksamhetskunskap för att kunna utforma informationssystemet. Verksamhetskunskap innefattar kunskaper om specifika förhållanden, om arbetssätt, om regler och rutiner och om mål av olika slag. Kunskap om systemutveckling innefattar kunskap om utrustning, om programvaror, om metoder för analys och design samt om teorier och modeller för olika aspekter av utvecklingsarbetet. Beroende på komplexitet i det utvecklade systemet (och den information som skall formaliseras) krävs olika mycket kunskap. Båda typerna av kunskap innefattar element som kan upplevas som svåra att förvärva. Verksamhetskunskap kan t.ex. innefatta etiska aspekter som inte enkelt kan uttryckas i ord. Denna omständighet gäller även kunskap om utveckling av informationssystem. En väsentlig skillnad är dock att specialisten är till för beställaren/användaren och inte tvärtom. Specialisterna är normalt en del av verksamhetens servicefunktion.

Kunskapsöverföringsproblemet kan hanteras på olika sätt. Ett sätt är användande av metoder för systemutveckling, där närmandet mellan beställare/användare ges hög prioritet, t.ex. med s.k. participatoriska ansatser. Användarmedverkan i systemutveck-

lingsprocessen har poängterats i systemutvecklingsmetoder allt sedan Langefors först uppmärksammade värdet av detta. Ett annat sätt är att utbilda IT-specialisten i verksamhetskunskap. I verksamheter där en IT-specialist arbetat länge kan en stor verksamhetskunskap utvecklas (se exempel i kap 5.2.5). Ytterligare ett sätt är att byta arbetsuppgifter i en verksamhet, s.k. arbetsrotation (Moore 1997), för att på så sätt få kunskaper om varandras arbetsförhållanden.

För att helt kunna eliminera kunskapsöverföringsproblemet krävs att beställaren/användaren kan utveckla sitt system själv. För att kunna utveckla egna system förutsätts kunskaper om utveckling av informationssystem hos användaren. Dessutom krävs tillgång till någon programvara för utveckling (utvecklingsverktyg), som är tillräckligt lättillgänglig. Exempel på sådana programvaror är CAD-program, databashanteringsprogram och kalkylprogram. I denna avhandling är det kalkylprogram som systemutvecklingsverktyg för användare, som är i fokus. En ansats där användare utvecklar egna system är bara möjlig då systemet är av begränsad storlek och komplexitet. Ansatsen innebär för- och nackdelar. T.ex. kan det framstå som en tveksam prioritering att låta högt betalda beslutsfattare syssla med sådant som IT-specialister kan utföra på kortare tid. De exempel som redovisas i denna avhandling är i övervägande fall av en sådan karaktär att de inte skulle bli aktuella för TSU-projekt. Problem med anveckling behandlas mer utförligt i kapitel 10.

1.1.6 Kalkylprogram och kalkylsystem

Då någon skapar en kombination av program och data som normalt kan återanvändas på något sätt har ett informationssystem (IS) skapats.¹² Informationssystem är ett synnerligen centralt begrepp i IT-historien och utveckling av informationssystem är en mycket central verksamhet inom IT-området. Ett informationssystem är resultatet av ett utvecklingsarbete utfört av en eller flera personer som fungerar som systemutvecklare. För att utveckla ett informationssystem behövs ett program för utveckling och realisering¹³, som även det kan sägas vara ett informationssystem.

De första programmen för programutveckling (t.ex. assemblerer) var så konstruerade, att programmeraren för att kunna använda assemblern, var tvungen att förstå datorns logiska uppbyggnad. Att lära sig assemblerspråk tillräckligt bra för att kunna skapa ett informationssystem var (och är) tidsödande, eftersom det krävde att man förutom goda insikter i datorns arbetssätt också har goda kunskaper i språkets syntax. Då s.k. tredje generationens språk (t.ex. Fortran och Cobol) blev tillgängliga krävdes inte samma djupa insikter i datorkunskap för att kunna skapa informationssystem. Och då s.k. fjärde generationens språk (t.ex. Access, dBase och Paradox) kunde användas krävdes ännu mindre dator- och syntaxkunskaper för att kunna utveckla mindre sy-

¹² Enligt den traditionella definitionen av informationssystem består det av funktioner för insamling, bearbetning, lagring, överföring och presentation av information (Andersen 1994:14f). Här behandlas endast datoriserade informationssystem. En mer utförlig diskussion av begreppet informationssystem genomförs i kapitel 8.

¹³ T.ex. en Cobolkompilator, en Basicinterpretator, ett CASE-verktyg eller ett kalkylprogram.

Introduktion

stem. Då denna avhandling skrivs (1999) är det möjligt att göra enkla tillämpningar med små¹⁴ kunskaper om datorers uppbyggnad och arbetssätt.

Persondatorer och *kalkylprogram* introducerades och blev vanliga på 80-talet. Näst ordbehandlingsprogram är kalkylprogram den vanligaste standardprogramvaran (Panko 1988, Ledell 1993, Nardi, Miller 1990). Som systemutvecklingsverktyg har kalkylprogram utnyttjats av användare. Personer som yrkesmässigt utvecklar informationssystem, använder normalt andra verktyg för systemutveckling. Uttrycket kalkyl kommer från latinets *calculus* och används oftast i betydelsen *beräkning*. Det engelska uttrycket för kalkylprogram är *spreadsheet program*. Termen spreadsheet syftar på rutnätet på kalkylbladet där användaren matar in värden, text och formler. Det svenska ordet *kalkyl* antyder vad programmet är tänkt att användas till, nämligen interaktiv¹⁵ beräkning. Denna användningsform (beräkning) är dock endast en av flera, om än den vanligaste. Det engelska uttrycket spreadsheet är mer neutralt till användningsformen och skulle kanske vara att föredra på svenska då *kalkylsystem*¹⁶ inte alltid innehåller någon form av beräkning. Men, eftersom benämningen kalkylprogram får anses vara väletablerad, får benämningen kalkylsystem även gälla för applikationer som inte innehåller egentliga kalkyler. Enligt Panko (1988:388) innehåller kalkylprogram fyra grundläggande funktioner, nämligen kalkylbladsanalys (spreadsheet analysis), grafik, databas och makro. Denna indelning följer programfunktionerna (se även kap 15.2).

Sammanfattning utveckling IT-området

Utvecklingen på IT-området har bl.a. resulterat i icke-procedurella systemutvecklingsverktyg, t.ex. kalkylprogram. Kalkylprogram är ett utvecklingsverktyg, som är tillgängligt för stora grupper verksamhetsföreträdare (användare). Kalkylprogram kan användas för systemutveckling utan kunskapsöverföringsproblem, dvs av verksamhetsföreträdarna själva, vilket innebär att det är ett alternativ till traditionell systemutveckling där kunskapsöverföringsproblemet existerar.

1.2 Kalkylprogramanveckling

Då kalkylprogram används av verksamhetsföreträdare för att självständigt tillgodose informationsbehov benämns detta i denna avhandling som *kalkylprogramanveckling*¹⁷. Begreppet kalkylprogramanveckling (KPA) är synonymt med *arbetsintegrerad systemutveckling med kalkylprogram* (AIS-K) (Avdic 1995a). KPA är ett specialfall av *anveckling*¹⁸. I detta avsnitt introduceras anvecklingsbegreppet utifrån en diskussion om användning och utveckling och relationerna däremellan. Begreppen *anveckling* och *kalkylprogramanveckling* definieras på sid 24.

¹⁴ Kunskapskraven minskar från år till år.

¹⁵ "**Interaktiv** ...I databehandlingssammanhang kallas program interaktiva då människa och dator på ett dialogliknande sätt växelvis bidrar till att den önskade uppgiften utförs." (Nationalencyklopedin)

¹⁶ Informationssystem utvecklade med kalkylprogram och där kalkylprogrammet fungerar som en del av systemet. Se vidare kapitel 8.3.

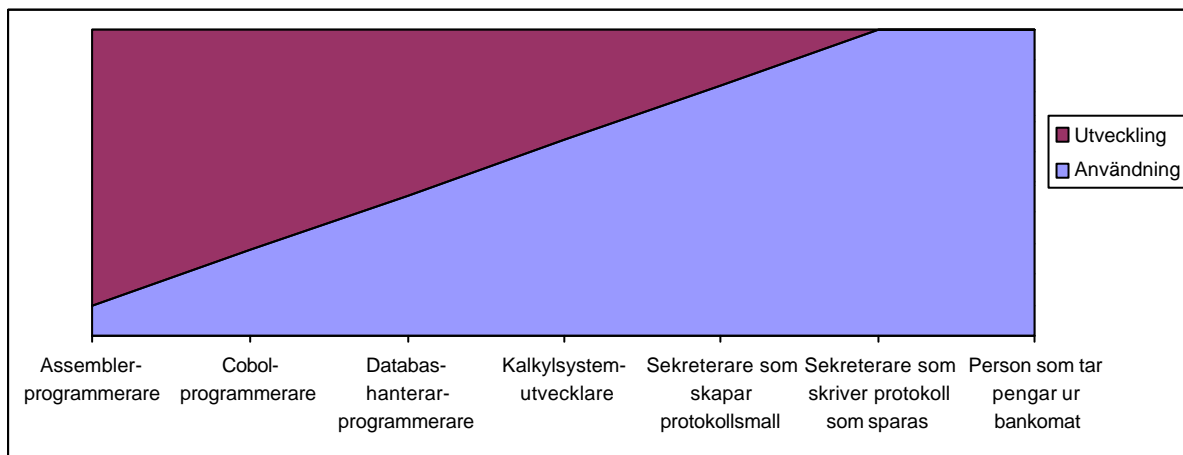
¹⁷ Termen kommer från uttrycket *användarutveckling med kalkylprogram*. Motiv till val av term diskuteras nedan.

¹⁸ Termen kommer från uttrycket *användarutveckling*. Motiv till val av term diskuteras nedan.

1.2.1 Relation mellan användning och utveckling

Alla som använder datorer är i någon mån informationssystemanvändare och åtskilliga är i någon mån systemutvecklare.¹⁹ En programmerare som skriver ett program som skall ingå i ett informationssystem är t.ex. användare av såväl kompilator som operativsystem (direkt och indirekt via kompilatorn) och han är förstås systemutvecklare eftersom han utvecklar ett system. En sekreterare som tillverkar en protokollsmall för användning av andra sekreterare är även han i denna bemärkelse systemutvecklare då han skapar datorstödda rutiner (informationssystem). Om däremot en sekreterare som skriver ett protokoll (med eller utan protokollsmall), som skall distribueras enligt en sändlista är han ju användare av ordbehandlingsprogrammet han skriver med, men knappast utvecklare.

Systemutveckling sker enligt ovanstående resonemang, då en person skapar formaliserade (enkla eller komplexa) datorstödda rutiner med hjälp av ett datoriserat utvecklingsverktyg, t.ex. en Cobolkompilator eller ett kalkylprogram. Användning sker då en person använder det utvecklade systemet. På så vis kan det anföras att systemutveckling normalt också innebär användning. En systemutvecklare som inte är användare av ett annat system skulle i så fall vara en som bygger datorn inklusive operativsystemet utan andra hjälpmedel än rent mekaniska. Exempel på sådana personer skulle kunna vara ENIAC:s konstruktörer Eckert och Mauchly. Idag är sådan aktivitet ovanlig om den finns överhuvudtaget, eftersom datorer används vid konstruktion av datorer, t.ex. för optimering av processorer. I Figur 6 nedan beskrivs schematiskt hur tid för utveckling och användning kan tänkas fördela sig för olika kategorier av datoranvändare och hur denna tid schematiskt



Figur 6 Schematisk beskrivning av relation mellan användning och utveckling av informationssystem hos datoranvändarkategorier.

fördelar sig mellan användning och utveckling utan att göra anspråk på att presentera några korrekta värden. Figuren syftar mera till att peka på ett principförhållande när det gäller datoranvändning än att kartlägga specifika grupperns situation. Assemblerprogrammeraren representerar följaktligen en datoranvändningsform där en stor del av

¹⁹ Förutsatt att man definierar informationssystem som innefattande rutiner i anslutning till systemet som finns för att systemets pragmatiska syfte skall uppfyllas. Begreppet informationssystem diskuteras mer ingående i kapitel 8.

Introduktion

tiden går åt till att utveckla informationssystem och person som använder bankomat representerar motsatsen.

Alla utvecklare sysslar med både utveckling och användning, medan vissa användargrupper enbart ägnar sig åt användning. Figuren visar olika exempel på kombinationer av användning och utveckling. I figuren placeras sekreterare som skriver protokoll som sparas och används till vänster om personer som tar pengar ur bankomat. Skälet till detta är att det skapade protokollet i någon mening kan tänkas representera en del av ett manuellt informationssystem. Att ta pengar ur bankomat framstår enligt denna indelning som den mest utpräglade formen av användning.

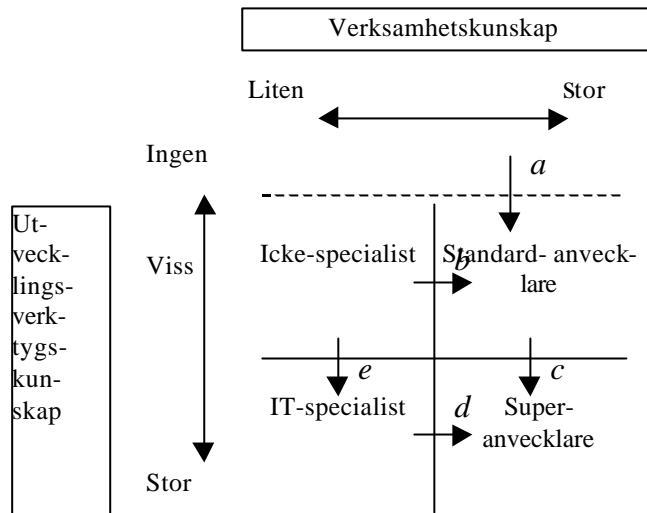
Figuren syftar till att visa att med utvecklingsverktyg av typen kalkylprogram, uppstår alltmer en gråzon mellan utveckling och användning som inte låter sig klassificeras enligt den uppdelning som var möjlig då TSU var den dominerande utvecklingsformen. Kategorier i mitten av figuren, kan sägas vara mer användare än utvecklare. De är i allmänhet inte utvecklare i den traditionella meningen som innebär att man använder ett utvecklingsverktyg av typ C-kompilator e.d. för att utveckla system till användare. Det behöver dock inte vara den använda programvaran som styr huruvida man räknas som utvecklare eller användare. Traditionell systemutveckling kan i princip utföras även med t.ex. kalkylprogram om utvecklaren använder programvaran som ett renodlat utvecklingsverktyg och utvecklar system till en beställare.

1.2.2 Anvecklare

Det som avgör huruvida man är en utvecklare, användare eller anvecklare, enligt den indelning som ligger till grund för denna avhandling, är om man har *verksamhetskunskap* (se även kap 13.1) eller ej i förhållande till den uppgift som skall utföras. Att tala om utveckling, användning eller anveckling utan att relatera till verksamhetskunskap framstår enligt detta resonemang som omöjligt. Förhållandena mellan olika kunskaps typer och roller åskådliggörs i Figur 7 nedan där fyra typer av kunskapskombinationer identifieras schematiskt. En person med liten verksamhetskunskap och viss kunskap om utvecklingsverktyg²⁰ (här kallad *icke-specialist*) kan tänkas fungera som utförare av rutinmässiga uppgifter med eller utan datorinslag. En person med liten verksamhetskunskap och stor kunskap om utvecklingsverktyg (här kallad *IT-specialist*) kan inte vara en anvecklare annat än i relation till sitt eget verksamhetsområde, i relation till utvecklingsverktyget. Exempel på IT-specialist kan man t.ex. hitta på dataavdelningar där det kan finnas personer med stora kunskaper om nätverk, C++, skrivarkonfiguration etc. Detta innebär inte att alla personer på dataavdelningar måste vara IT-specialister, även om de flesta förmodligen är det i något avseende.

²⁰ Med kunskap om utvecklingsverktyg avses här en delmängd av kunskap om systemutveckling. I resonemanget avses främst kunskap om programvaror med vars hjälp informationssystem kan utvecklas och realiserar.

Användare och utvecklare



Figur 7 Schematisk beskrivning av förhållandet mellan verksamhetskunskap och kunskap om utvecklingsverktyg.

En person med stor verksamhetskunskap och liten, men någon, kunskap om utvecklingsverktyg (här kallad *standardanvecklare*) kan sägas vara den typiske anvecklaren. Exempel på en standardanvecklare skulle kunna vara en ekonom som utvecklar en budgetmodell i sitt arbete. Ju mer kunskap om utvecklingsverktyg, desto mindre är problemet med kunskapsöverföring vid systemutvecklingsarbete. En person med stor verksamhetskunskap och stor kunskap om utvecklingsverktyg (här kallad *superanvecklare*) är också en anvecklare eftersom han/hon kan utveckla komplexa system utan att behöva överföra kunskap till en specialist. Exempel på en superanvecklare skulle kunna vara en tekniker, som för eget bruk eller till sin avdelning, utvecklar mätsystem i C, Fortran eller något annat programmeringsverktyg. Ett annat exempel skulle kunna vara en systemutvecklare som utvecklar sitt eget programmeringsverktyg.

Kategoriindelningen i Figur 7 ovan är inte statisk. Personer med stor verksamhetskunskap kan bli anvecklare genom att tillägna sig vissa kunskaper om utvecklingsverktyg, t.ex. kalkylprogram (*a*). Genom lärande om verksamhet kan utveckling från icke-specialist till standardanvecklare (*b*) eller från standardanvecklare till superanvecklare (*c*) ske. Huruvida en person kan klassificeras som IT-specialist eller superanvecklare beror på vilken verksamhet som personen ifråga bedriver. Om personen har stora utvecklingsverktygskunskaper och utvecklar åt andra hamnar han/hon i IT-specialistrutan men om han/hon utvecklar för sin egen verksamhets räkning hamnar han/hon i superanvecklarrutan. En inte ovanlig situation är den att IT-specialister under den tid man arbetar på ett företag utvecklar verksamhetskunskaper och tenderar till att närma sig superanvecklarrutan (*d*). För att helt och hållet ingå i den rutan krävs dock att personen har arbetsuppgifterna i en verksamhet/delverksamhet som sin huvuduppgift och att han har (del-)ansvar för dessa arbetsuppgifter. IT-specialisten har alltid IT-verksamhet som sin huvuduppgift. I verkligheten kan gränsfall finnas. Den fjärde pilen (*e*) visar att en utveckling från icke-specialist till IT-specialist också är möjlig, fast inte intressant i ett KPA-sammanhang. Kategoriseringen i figuren är principiell och syftar till att peka ut innehav av stor verksamhetskunskap som en avgörande faktor för hur

systemutveckling kan bedrivas. Anveckling kan även ske grupporienterat. Ett arbetslag kan t.ex. utveckla kalkylsystem för sin egen verksamhets räkning (Avdic 1995a:78). Dels så kan själva konstruktionsarbetet utföras av flera personer och dels så kan en person sköta utvecklingen medan övriga använder systemet. Det som gör att det inte är frågan om TSU är att utvecklaren har arbetsuppgifter i verksamheten som sin huvuduppgift.

Preliminär definition anvecklare

Kriterier för att en person kan betraktas som anvecklare är att personen har stor verksamhetskunskap och viss kunskap om utvecklingsverktyg. Dessutom skall personen ha arbetsuppgifter i den verksamhet, som är föremål för anveckling, som sin huvuduppgift (se även kap 11.4).

1.2.3 Kunskap om utvecklingsverktyg

Med utvecklingsverktyg avses här hjälpmedel för utveckling i vid mening. Detta omfattar utvecklingsprogramvara, utrustning och metoder för analys och design av informationssystem. Kunskap om utvecklingsverktyg behöver utvecklaren för att utveckla datoriserade informationssystem utan att ta hjälp av IT-specialister. Kunskap om kalkylprogram är ett exempel på denna kunskapsstyp.

Kunskap om utvecklingsverktyg kan vara olika lättillgänglig. Vissa verktyg är primärt avsedda att användas för utveckling av stora och komplexa system. Programvaror för utveckling är ofta av procedurell karaktär (t.ex. en C-kompilator), vilket innebär att rutiner måste formaliseras i form av kod. Även om det finns hjälpmedel för att skapa en del av koden måste utvecklaren ha god förmåga till förståelse av programkod och kunskap om hanteringen av denna. Detta faktum leder till att utvecklare företrädesvis använder sig av icke-procedurella verktyg t.ex. kalkylprogram. De icke-procedurella verktygen kan variera avseende formaliseringssätt, dvs hur regler i systemet skapas av utvecklaren. Ett sätt att dela in funktionerna i verktygen är i typerna *deklarativa* och *direktmanipulerande*. Deklarativa funktioner innebär att man deklarerar i form av sats hur data skall struktureras, hur inmatningsformulär skall se ut etc. Direktmanipulerande verktyg är utpekande, vilket innebär mer att man efterliknar något så nära som möjligt. När man t.ex. skapar en formel i ett kalkylprogram kan man utföra detta genom att peka på ett värde i en cell och klicka. Det direktmanipulerande inslaget kan gälla för olika nivåer. Att peka och klicka på celler som innehåller värden med motsvarigheter i sinnevärlden t.ex. ett saldovärde pekar ut representationer av verksamheten medan t.ex. direktmanipulerande design av ett formulär för inmatning innebär utpekande av komponenter i ett gränssnitt. Verktyg kan ha inslag av såväl direktmanipulering²¹ som deklarativitet. Deklarativitet och direktmanipulering är två bland flera exempel på interaktionsformer. Andra interaktionsformer som finns är kommandospråk och menyer. Begreppet verktyg diskuteras utförligare i kapitel 15.

²¹ Med *direktmanipulering* avses interaktion med objekt snarare än med andra system (Smithson & Hirschheim 1990:37ff).

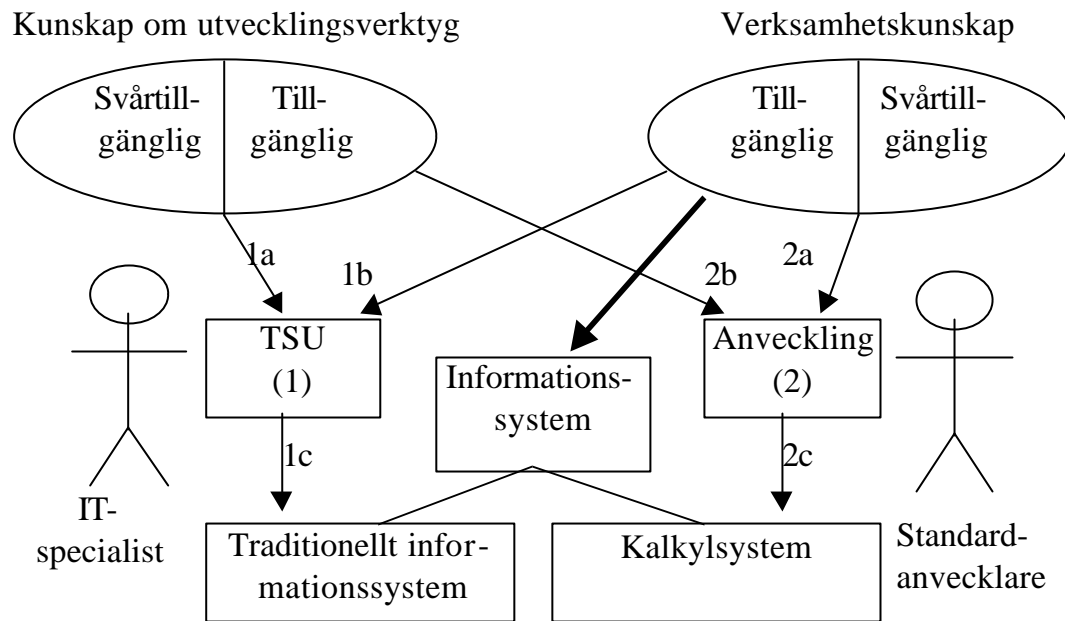
Utvecklingsverktyg kan även omfatta andra programvaror än de som används för den direkta utvecklingen. Programvaror för kommunikation och informationssökning, t.ex. Internet är ett exempel på detta. Utrustning av olika slag kan också räknas in under utvecklingsverktyg. Exempel på detta är datorer, skrivare, scanners etc. Även metoder kan räknas in som utvecklingsverktyg. Metoder för problemlösning och datastrukturering kan nämnas som exempel på detta.

1.2.4 Verksamhetskunskap

Enligt ovanstående resonemang är det frågan om huruvida en person har verksamhetskunskap respektive kunskap om utvecklingsverktyg, som avgör hur kunskapsöverföringsproblemet kan hanteras. Med verksamhetskunskap avses den sammansättning av olika kunskaper om en given verksamhet, som en person behöver för att kunna utföra en given uppgift i denna verksamhet. Verksamhetskunskap består, som tidigare berörts, av olika typer av kunskaper. Det kan t.ex. vara faktakunskaper, färdighetskunskaper eller kunskaper om regler och mål.²² Olika typer av kunskaper kan vara olika svåra att kommunicera och formalisera. Därmed kan kunskap ibland också vara svår att överföra t.ex. till en systemutvecklare. Verksamhetskunskap har alla. En snickare eller en arborrare har det liksom en arkitekt eller en lärare. Även IT-specialister (t.ex. systemutvecklare och programmerare) har verksamhetskunskap. Det speciella med IT-specialisternas verksamhetskunskap är att den i flera avseenden sammanfaller med kunskaper om utvecklingsverktyg.

Förhållandet mellan kunskapstyper och utvecklingstyper beskrivs schematiskt i Figur 8 nedan. Figuren syftar till att visa skillnaden mellan TSU och anveckling vad avser behov av kunskapstyper. Vid traditionell systemutveckling (1) har IT-specialisten kunskap om utvecklingsverktyg (1a) och behöver tillgång till verksamhetskunskap (1b) för att utveckla. Anvecklaren utgår i sitt anvecklande (2) från sin verksamhetskunskap och behöver kunskap om utvecklingsverktyg (2b) för att skapa t.ex. ett kalkylsystem (2c). Noteras kan också att IT-specialisten har tillgång till mer kunskaper om utvecklingsverktyg än standardanvecklaren, samt att standardanvecklaren har tillgång till mer verksamhetskunskap än IT-specialisten. Den tjocka linjen mitt i figuren syftar till att visa att informationssystem, såväl kalkylsystem som TIS, handlar om verksamhetskunskap. Detta konstaterande innebär att verksamhetskunskap är överordnad kunskap om utvecklingsverktyg.

²² Analysen av begreppet verksamhetskunskap fördjupas i kapitel 13.1. Där görs t.ex. viss åtskillnad mellan normer och kunskap. Verksamhetskunskap fungerar här som en övergripande benämning.



Figur 8 Schematisk beskrivning av förhållandet mellan kunskap och utveckling.

1.2.5 Verksamhetsmotverkande chauvinism

En utgångspunkt för denna avhandling är att anvecklaren är den som normalt bäst kan ta ställning till vad som är den lämpligaste åtgärden i anvecklarens verksamhet. Speciellt gäller detta jämfört med IT-specialister. Skälet till denna utgångspunkt utvecklas nedan.

Ståndpunkten att anvecklaren är den som vet bäst är inte självklar. Ståndpunkten är på sätt och vis uttryck för ett harmoniperspektiv som kan förefalla något orealistiskt.²³ Det är t.ex. inte rimligt att utgå från att företag och andra organisationer är fria från motsättningar och suboptimering av olika slag där anvecklaren kan tänkas agera i eget intresse och i motsats till andra aktörer i organisationen. Tillvaron är fylld med exempel på intressemotsättningar av olika karaktär. I en organisation kan detta leda till vad som här skulle kunna kallas *verksamhetsmotverkande chauvinism*²⁴. Verksamhetsmotverkande är inget absolut begrepp. Det avser att poängtera att det i de allra flesta fall går att säga att ett givet agerande från en person, kan betecknas som i enlighet med vad verksamheten och/eller samhället strävar mot. Här kan dock anföras mängder med gränsfall. Några exempel följer nedan:

- Anvecklaren skulle av "fackliga" skäl kunna avstå från att agera utvecklare. "Det är inte mitt jobb".
- Anvecklaren skulle kunna utveckla kalkylsystem för att underminera dataavdelningens inflytande. "Dom där vill jag inte vara beroende av".

²³ Med harmoniperspektiv avses här ett synsätt som ser på aktörer i ett företag som strävande mot ett och samma mål. Synsättet står i motsats till ett dialektiskt synsätt, som ser motsättningar som en naturlig och inneboende del av verksamheter.

²⁴ Chauvinism = "...överdriven partiskhet för eller uppskattning av viss grupp eller plats; den kan även beskriva överlägsen attityd gentemot andra..." (Nationalencyklopedin).

Användare och utvecklare

- Anvecklaren skulle kunna utveckla kalkylsystem som gjorde företaget beroende av honom. "Då kan jag inte få sparken".
- Anvecklaren skulle kunna syssla med anveckling för att det var kul och strunta i om det var nyttigt för företaget. "Tjoho!"
- Anvecklaren skulle kunna utveckla kalkylsystem som fryser fast föråldrade och suboptimerade arbetssätt. "Jag vet bäst!"

Att utgå från att anvecklaren vet bäst, och att inte ta med fenomenet verksamhetsmotverkande chauvinism då anveckling studeras, innebär det att man bortser från fenomenet? Svaret är nej och argumenten för detta redovisas nedan.

Avhandlingen utgår från en syn på datorn och dess program som verktyg, jämförbara med andra verktyg, som kan tänkas finnas på en arbetsplats. Med verktyg avses här hjälpmedel i största allmänhet. Finns det då något som motiverar att man behandlar kalkylprogram (eller anvecklarprogram i allmänhet) som speciella jämfört med andra hjälpmedel? Finns det något som gör att just kalkylprogram skapar speciella problem jämfört med andra faciliteter i en administrativ miljö? En utgångspunkt för avhandlingen är att de rutiner, som finns i en verksamhet vad gäller att arbeta mot verksamhetens mål, gäller för anveckling på samma sätt som det gäller för kreativt utvecklingsarbete i största allmänhet. Om t.ex. en ekonom har i uppgift att utforma budgetrutiner, finns det då skäl att granska just de delar som omfattar anveckling med kalkylprogram speciellt? Kan man säga att annan utvecklingsverksamhet som rutinutformning, blankettutformning, fastställande av beräkningsmodeller, val av nyckeltal, fastställande av tidsgränser etc. skulle ha en kvalitativt annan karaktär än just anveckling? Frågorna är kategoriskt ställda och kan leda till två principiella svar. Antingen 1) Ja, anveckling är speciell eller 2) Nej, samma gäller för allt utvecklingsarbete.

Om man utgår från alternativ 1, så uppstår frågan varför anveckling skulle vara anorlunda än annan utveckling? Varför skulle t.ex. ett budgetsystem utvecklat i kalkylprogrammiljö vara mer utsatt för (rationell eller icke-rationell) verksamhetsmotverkande chauvinism än ett manuellt budgetsystem. En orsak skulle kunna vara att personberoende blir större om det bygger på programvarukunskap. Ett annat (möjligen mindre rimligt) skäl till att anveckling kan framstå som en speciell form av utveckling är att det traditionellt utvecklats system av specialister från dataavdelningen. Denna tradition gör kanhända att anvecklares system kan vara svårare att acceptera än dataavdelningens. Denna invändning skulle i så fall bygga på att dataavdelningen kan anses ha större möjligheter och/eller kunskaper ägnade att utveckla lämpliga system än anvecklare. Risker för verksamhetsmotverkande chauvinism får dock anses gälla även för dataavdelningens personal. Denna egenskap kan innehas av alla personer i en organisation, inklusive personer i organisationens ledning. Egenskapen kan även gälla personer i det vetenskapliga samhället, i media eller i samhällsledningen.

Om alternativ 2 beaktas, så kan anföras att undanröjande av verksamhetsmotverkande chauvinism får anses vara något som en organisation har att hantera i största allmänhet oberoende av om datorer är inblandade eller ej. Avhandlingens utgångspunkt är att anvecklaren i normalfallet själv strävar efter att göra kalkylsystemen genomskinliga

Introduktion

och så personoberoende som möjligt. Om anvecklaren inte strävar efter detta kan det förefalla sannolikt att han agerar för att göra sig outhärlig även i andra sammanhang än när det gäller anveckling. Det är här som diskussionen om yrkes- och verksamhetsetik kommer in (se nedan). I normalfallet verkar och finns anvecklaren dessutom i en professionell miljö där arbetsledning, kollegor och andra intressenter påverkar beslutsfattande och dess förutsättningar.

Ytterligare ett argument för att anvecklaren framstår som den som är mest lämpad att veta vad som är den lämpligaste åtgärden i anveklarens verksamhet är att anveklarna helt eller delvis ansvarar för de verksamheter som de färdiga kalkylsystemen avser att betjäna. Detta leder till att det är anvecklaren, som får bära ansvaret för eventuella negativa konsekvenser av anvecklingen.

Sammanfattning verksamhetsmotverkande chauvinism

Anveckling är inte att betrakta som annorlunda än annan utvecklande verksamhet och faller därmed under det uttalade eller outtalade regelverk, som förutsätts finnas inom olika professioner. Om detta inte är fallet, förutsätts att de normala ledningsfunktionernas kvalitetssäkrande verksamhet kommer att hantera verksamhetsmotverkande chauvinism såväl inom anveckling, som inom annan utvecklande verksamhet. Anveklarens verksamhetsansvar bidrar också till att anvecklaren normalt bäst kan ta ställning till vad som är den lämpligaste åtgärden i anveklarens verksamhet, förutsatt att behov för åtgärder anses finnas.

(Resonemanget är inget hinder för att åtgärder riktade mot verksamhetsmotverkande chauvinism skulle kunna vidtagas i de fall som det är tillämpligt på anveckling. Ett exempel skulle kunna vara att överväga riktlinjer för att motverka personberoende anveckling. Detta skulle t.ex. kunna ske genom att betona vikten av att använda namn istället för cellreferenser i formler för att logiken i ett kalkylsystem skall bli mer genomskinlig.)

1.2.6 Yrkesetik

En viss typ av svårformaliserbar verksamhetskunskap är *yrkesetik*. (Avdic 1995a:68) Begreppet yrkesetik står i detta sammanhang för (moraliska) normer, skrivna eller oskrivna, som spelar en viktig roll i ett yrke eller i en viss arbetsuppgift (Johnson 1985:7). Varje yrkesgrupp har sin egen uppsättning normer även om de kan ha en del av dessa normer gemensamma med andra yrkesgrupper (Collste 1996:78). Normerna består av krav på beteende vad gäller relationer mot andra personer, grupper av personer eller organisationer. Det kan gälla sådant som man uppfattar som kollegors, arbetsgivares, fackföreningars eller samhällets uttalade eller outtalade krav på hur man skall agera eller inte agera. Etiska normer kan betraktas som mer eller mindre underrespektive överordnade varann. De mest överordnade är av karaktären rättvisa, jämlikhet, oberoende, frihet etc. De underordnade normerna kan mycket väl stå i motsats till varandra och det kan vara svårt att veta vilka man skall prioritera. Moraliska konflikter kan uppstå. När informationssystem byggs är dessa etiska aspekter av intresse av det skälet att informationssystem är modeller av verkligheten och vare sig man vill eller

inte så byggs det in i systemen möjligheter att bättre eller sämre ta tillvara dessa etiska aspekter. Att uppfylla de mål som dessa normer syftar till kan uppfattas som centralt i olika yrken eller sysselsättningar. Avdic (1995a) visar att KPA kan bidra till att dessa yrkesetiska normer kan uppfyllas. Exempel på uttalade yrkesetiska normer är FAR:s, (Föreningen Auktoriserade Revisorers) regler. Ett annat exempel är ACM:s (Association of Computing Machinery) regler (se Bilaga 5).

1.2.7 Verksamhetsetik

Hittillsvarande resonemang antyder att det kan finnas åtminstone två typer av *svårformaliserbar kunskap* i form av etik (se även kap 13.4). Å ena sidan kan man som nämnts tala om kunskap knuten till ett yrke, yrkesetik och å andra sidan kunskap knuten till en specifik verksamhet, *verksamhetsetik*. Givet en viss yrkesgrupp, t.ex. ekonomer, så kan gruppens yrkesetik anses gälla oberoende av i vilken verksamhet ekonomen verkar. På samma sätt kan det i en given verksamhet, t.ex. en byggnadsfirma, tänkas finnas regler och värderingar relaterade till verksamheten, som gäller för i verksamheten verkande yrkesgrupper, såväl ingenjörer som ekonomer och övriga. Exempel på detta skulle kunna vara att ett allmännyttigt bostadsföretag strävar efter att förse alla samhällsgrupper i en kommun med billigast möjliga boende av en viss standard (Avdic 1995a:67ff). Andra exempel som är verksamhetsrelaterade skulle kunna vara ett byggföretag som strävar efter att endast använda återvinningsbara råvaror eller att en högskola strävar efter att endast examinera konstruktivt kritiska studenter. På samma sätt som när det gäller yrkesetik finns det över- och underordnade regler. De överordnade reglerna kan vara av samma karaktär som gällde för yrkesetik, medan de underordnade kan vara relaterade till den specifika verksamheten.

1.2.8 Yrkes- och verksamhetsetik

De båda etiktyperna kan vara överlappande och samverkande. Tillsammans kan de sägas utgöra *yrkes- och verksamhetsetik*. Att just de etiska aspekterna framhålls beror på att de kan representera de mål, som anvecklaren har att styra mot i sina arbetsuppgifter. Kunskap om ovan beskrivna etiktyper, kan sägas vara svårformaliserbar på så vis att den inte enkelt låter sig kvantifieras och uttryckas som formler eller funktioner och därmed inte heller låter sig implementeras i datoriserade informationssystem annat än möjligen indirekt via systemets syfte.

Alltsedan datorns tillkomst har det funnits ambitioner att formalisera mer och mer av mänskliga funktioner och mänsklig verksamhet. De tidiga systemen automatiserade rent matematiska förhållanden, som redan var uttryckta kvantitativt i någon form. Allteftersom datorer har utvecklats och förfinats har allt mer avancerade ambitioner sett dagens ljus. Exempel på detta är området AI (Artificiell Intelligens), som under en period förutsågs kunna automatisera sådana delar av människans intellektuella funktioner, som inte enkelt, om alls, låter sig uttryckas kvantitativt. I Japan, USA och Europa satsades under 1980-talet avsevärda resurser på AI. Kanske har förhoppningarna dämpats något vad gäller AI:s möjligheter. Svårigheterna att formalisera icke-kvantitativa fenomen, t.ex. vissa mål i verksamheter och yrken, är inte triviala. Kunskap om yrkes-

Introduktion

och verksamhetsetik kan när den är svårformaliserbar räknas in under benämningen *tyst kunskap* (se även kap 13.6). Tyst kunskap innefattar även färdigheter, som inte enkelt låter sig formaliseras eller kommuniceras. Exempel på detta kan vara hantverkskunskaper av olika slag.

Mot bakgrund av diskussionerna om olika typer av kunskap och svårigheterna att formalisera och kommunicera icke-kvantitativa fenomen framstår anveckling som ett tänkbart sätt att kunna utnyttja datorkraft och ta hänsyn till svårformaliserbar kunskap utan att behöva formalisera den. Om anvecklare har tillgång till programvara för utveckling av informationssystem, som är tillräckligt kraftfull för att i vissa fall ersätta traditionell systemutveckling, elimineras kunskapsöverföringsproblemet. Det kan vidare tyckas rimligt att större hänsyn kan tas till kunskap, som är svår att formalisera och därmed uttrycka och kommunicera. I anveklarens kunskapsarsenal finns en inte alltid uttalad insikt om såväl yrkets som verksamhetens mål. Detta kommer till uttryck i den verksamhet som anveklaren bedriver i sin dagliga gärning. När anveklaren får ett utvecklingsverktyg tillgängligt, kan tyckas troligt att denna insikt kan få genomslag även där. Anvecklingsverksamheten är ju en del av anveklarens ordinarie verksamhet, med det kännetecknet att det tidigare inte varit möjligt för anveklaren att skapa datoriserade informationssystem utan hjälp av systemutvecklingsspecialister.²⁵

Sammanfattning Anveckling och yrkes- och verksamhetsetik

När verksamhetskunskaper och kunskaper om utvecklingsverktyg kombineras kan större hänsyn tas till verksamhetskunskaper i form av svårformaliserbar kunskap, t.ex. om yrkes- och verksamhetsetik.

1.2.9 Uttrycket Anveckling

För att beskriva fenomenet, som fokuseras i denna avhandling används uttrycket *anveckling* och *kalkylprogramanveckling* (KPA). Uttrycket anveckling är en förkortning av termen användarutveckling²⁶, som återfinns hos Lally (1995:3) och Hicks Jr (1990:192). Begreppet associerar till den etablerade benämningen på den grupp personer som förknippas med rollen användare av informationssystem (användare) samt benämningen på den aktivitet som resulterar i informationssystem (systemutveckling). Uttrycket användare innefattar betydelser, som är missvisande för den roll som de aktuella personerna utför samtidigt som benämningen är utomordentligt vanligt förekommande (se även kap 11). Av dessa skäl uppstår ett behov av ett uttryck som endast vagt refererar till uttrycket användare.

En fördel med uttrycket *anveklare* är att det är relativt kort och att det endast vagt associerar till ordet användare, vilket är en viktig aspekt. Den roll som uttrycket skall referera till är inte en användarroll i den mening, som traditionellt förknippas med ut-

²⁵ Denna ståndpunkt innebär inte att behovet av systemutvecklingsspecialister avskaffas, möjligen att det kan minskas i vissa avseenden, men även att det kan öka i andra avseenden. Om anveklaren blir mer medveten om möjligheter med IT, kan t.ex. behov av databaser eller processororientering av centrala informationssystem bli aktuella.

²⁶ "Userdevelopment".

trycket användare och vilket skulle vara missvisande och ge fel perspektiv. Ytterligare en aspekt är att uttrycket kan associeras såväl till aktivitet (anveckling & kalkylprogramanveckling) som till person (anvecklare). Denna fördel gäller inte för kombinationen användarutvecklare, användarutveckling och kalkylprogramanvändarutveckling eftersom det resulterar i långa uttryck (kalkylprogramanvändarutvecklare!). Uttrycket användarutveckling kan dessutom missvisande tolkas som att det är användare som skall utvecklas. Även *självbetjäning* (Nilsson 1991:59), som är ett tänkbart uttryck blir inte okomplicerat att ”rollifiera” (självbetjäna). Arbetsintegrerad systemutveckling med kalkylprogram (Avdic 1995a) är väl långt och har ingen naturlig rollform (skulle i så fall vara användarutvecklare). Uttrycket anvecklare avser att ge en avlägsen, ej alltför uttrycklig association till termen användare genom prefixet an. Slutligen är ordet anvecklare ett nytt ord, vilket kan passa bra med tanke på att det är fråga om ett relativt nytt fenomen. Termen är både ny och associerad till relaterade områden.

En nackdel med uttrycket skulle kunna vara att det låter lite ”konstigt” och därmed kan ge upphov till felaktiga associationer. Ytterligare en nackdel skulle kunna vara att uttrycket helt enkelt är onödigt, redan befintliga termer kan utnyttjats till att referera till önskat begrepp. En sista nackdel är att det inte finns en självklar engelsk översättning (a useloper uselopes? a develuser develuses?).

Sammanfattningsvis uppfattas fördelarna som väsentligare än nackdelarna, eftersom ett befintligt och hanterligt uttryck, som refererar till avsett begrepp, saknas.

Anveckling definieras i denna avhandling enligt följande:²⁷

Med ***anveckling*** avses en form av systemutveckling som initieras, drivs och/eller utförs av människor, som själva huvudsakligen fungerar som användare i någon mening. Användare används här i den traditionella meningen 'användare av datoriserade informationssystem'. Personerna, som utför utvecklingen (anvecklingen) kallas *anvecklare*. Den primära utgångspunkten är att arbetet bedrivs utifrån stora verksamhetskunskaper, d.v.s. kunskaper om arbetsuppgifter och behov förknippade med dessa. Arbetet bedrivs med utvecklingsverktyg (t.ex. kalkylprogram), som anvecklarna har kunskaper om. Anvecklarna ansvarar helt eller delvis för de verksamheter som det färdiga systemet avser att stödja. De utför även normalt arbetsuppgifter inom denna verksamhet. Då anveckling bedrivs med kalkylprogram kallas detta för *kalkylprogramanveckling* (KPA).

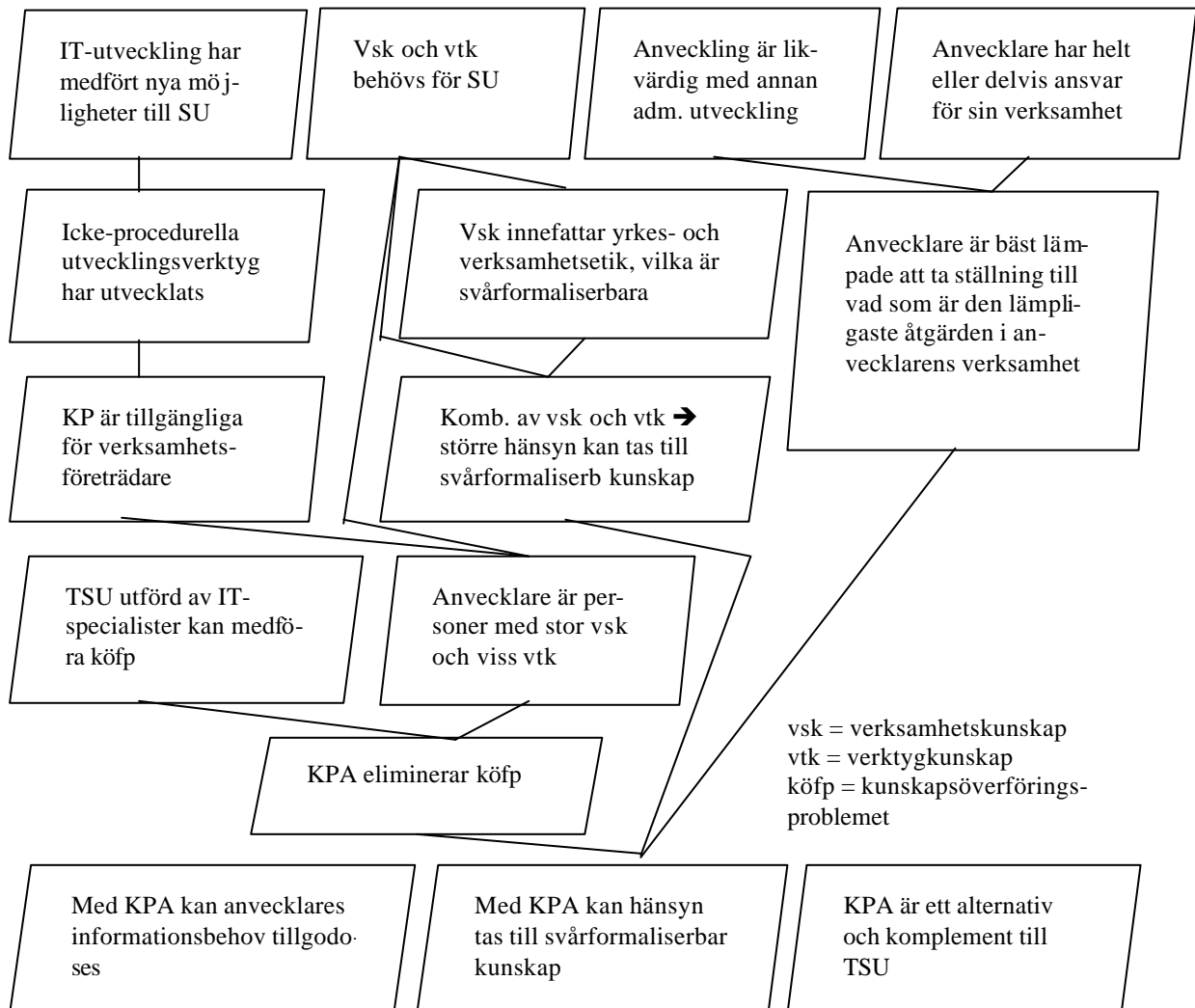
1.3 Forskningsfråga

Det fenomen som är föremål för studium i denna avhandling har sin upprinnelse i den förändring av förhållandet mellan användning och utveckling av informationssystem, som skett under senare år. Detta förhållande har belysts i de inledande två avsnitten ovan och skall nedan sammanfattas och preciseras.

²⁷ Definitionen är en bearbetning av Avdic (1995a).

1.3.1 Preliminära förutsättningar för kalkylprogramanveckling

Som antytts ovan finns det väsentliga skillnader mellan TSU och KPA. En av orsakerna till skillnaderna kan vara utvecklingen inom IT-området vad gäller bl.a. kostnadsminskning, prestandaökning och kunskaps-spridning (Avdic 1995a:198). Detta har lett till större möjligheter för människor att hantera datorer självständigt. Ett exempel på detta är systemutveckling med kalkylprogram, som kan fungera både som utvecklingsverktyg och slutanvändarhjälpmedel och som innebär att personer på egen hand kan bygga mer eller mindre komplexa informationssystem utan andra kunskaper än kännedom om sin verksamhet och sitt kalkylprogram. Detta förhållande kan å ena sidan förefalla intressant som ett alternativ till traditionell systemutveckling. Å andra sidan kan det förefalla sannolikt att förhållandet skulle föra med sig oönskade konsekvenser, som t.ex. icke-flexibla och odokumenterade system med dålig prestanda.



Figur 9 Kalkylprogramanveckling: Preliminära förutsättningar och innebörd.

Presentationen av avhandlingens forskningsfråga utgår från några tänkbara utgångspunkter för forskning kring kalkylprogramanveckling. En del av dessa utgångspunkter

har introducerats i detta inledande kapitel och en del har formulerats i Avdic (1995a). Nedan görs en sammanfattning av dessa utgångspunkter (se även Figur 9 ovan).

- Utvecklingen inom IT-området har gjort det möjligt för icke-IT-specialister att utforma informationssystem för att utföra egna arbetsuppgifter/tillgodose informationsbehov. Denna aktivitet kan benämnas anveckling. Ett specialfall av anveckling är kalkylprogramanveckling (KPA).
- Anveckling innebär en förändrad rollfördelning vid systemutveckling jämfört med traditionell systemutveckling (TSU) på så vis att verksamhetsföreträdare i form av beställare/användare antar rollen av utvecklare för att utföra sina arbetsuppgifter. Detta innebär att anveckling är ett alternativ till TSU.
- Kunskapsöverföring är ett problem vid TSU.
- Kombinationen verksamhetskunskap och kunskap om utvecklingsverktyg skapar förutsättning för anveckling och eliminerande av kunskapsöverföringsproblemet.
- Icke-procedurella utvecklingsverktyg, t.ex. kalkylprogram, har egenskaper som gör dem lämpliga för anveckling.

1.3.2 Tänkbara effekter

KPA innebär ett radikalt annorlunda sätt att bedriva systemutveckling jämfört med traditionell systemutveckling. Detta medför effekter som kan innebära möjligheter och problem. En tänkbar effekt av KPA utifrån ovanstående antaganden är att datorn alltmer kommer att utnyttjas som ett personligt verktyg. I begreppet verktyg ligger en aspekt att verktyget skall bli "osynligt", dvs att det används mer eller mindre intuitivt, ungefär som en snickare använder sin hammare. När datorn alltmer utvecklas till ett intuitivt verktyg kommer människors arbetsuppgifter att i högre grad kunna styra användningen av datorer jämfört med tidigare. Då datorn blir ett mer lättillgängligt personligt verktyg för personer kan följande effekter inträffa:

- Fler verksamhetsföreträdare kommer att bedriva KPA.
- KPA kommer att kunna stödja fler typer av arbetsuppgifter.
- KPA kommer att bli en naturlig del av vissa arbetsuppgifters utförande.
- Utvecklingsspecialister kommer att få delvis andra uppgifter och andra relationer gentemot anvecklarna i och med att dessa delvis agerar som utvecklare.
- Ansvar för systemutveckling och IT-utnyttjande kommer att förskjutas mot linjeverksamheten.²⁸
- Anvecklare kommer att utföra vissa traditionella IT-specialistuppgifter, vilket i sin tur kommer att leda till ett antal effekter, t.ex. nya organisationsstrukturer för administration av IT.
- Kalkylprogram kommer att kunna fungera som en länk till databaser i interna och externa miljöer, där man kan hämta och bearbeta information utan att ha djupare IT-specialistkunskaper.
- Möjligheter att utnyttja verksamhetskunskap och styra mot svårformaliserbara mål ökar då kunskapsöverföringsproblemet elimineras (Avdic 1995a:200).

²⁸ En annan tänkbar tendens är att utvecklingsspecialisterna som sådana kommer att "outsourcas" till linjeverksamheten, vilket indirekt kan tänkas ha visst inflytande på villkoren för KPA.

Avhandlingen bygger på antagandet att då man (med t.ex. KPA) eliminerar kunskapsöverföringsproblemet vid systemutveckling så kan verksamhetskunskaper utnyttjas på ett kvalitativt annorlunda sätt jämfört med TSU. Speciellt förväntas anvecklaren kunna ta hänsyn till svårformaliserbar kunskap av yrkes- och verksamhetsetisk art. För att detta skall vara möjligt krävs att anvecklaren har tillgång till och kunskap om utvecklingsverktyg. Ett exempel på ett sådant verktyg är kalkylprogram. Olika typer och nivåer av verktygskunskap ger olika möjligheter till anveckling. Hur detta sker och vilka effekter det får är studieområdet för avhandlingen.

1.3.3 Formulering av forskningsfrågor

Ovanstående översiktliga beskrivning av preliminära förutsättningar, innebörd och effekter av KPA grundar sig huvudsakligen på Avdic (1995a), vilken är att betrakta som en explorativ studie inom området. Föreliggande avhandling studerar samma område (KPA) med ambitionen att studera relationer mellan förutsättningar för och effekter av KPA. Med förutsättningar avses företeelser och/eller förhållanden som kan antas resultera i, för någon intressent, positiva eller negativa effekter. Speciellt beaktas möjligheten att förutom att uppfylla anvecklarens informationsbehov, även ta hänsyn till svårformaliserbar kunskap och uppfylla svårformaliserbara mål. För att förtydliga skulle tänkbara förutsättningar kunna exemplifieras med: olika kunskapsstyper, t.ex. kunskap om datamodellering eller frågespråk, tillgång till utrustning och programvara, t.ex. till ett intranät, färgskrivare eller viss version av någon programvara. På samma sätt skulle exempel på effekter kunna vara möjlighet till presentation och kommunikation eller möjlighet att ta sig an i något avseende komplexa uppgifter.²⁹ Avhandlingens tillkomst bottnar i en övertygelse om att möjligheterna ökar för verksamhetsföreträdare (även kallade användare) att utföra arbetsuppgifter då de kan använda kalkylprogram för att bygga informationssystem.

Den övergripande forskningsfrågan för denna avhandling kan formuleras:

Vilka nya möjligheter får användare att utföra arbetsuppgifter då de själva kan bygga informationssystem?

Traditionell systemutveckling bygger på att systemutvecklaren (IT-specialisten) utgår från sin utvecklingsverktygskunskap och tillägnar sig verksamhetskunskap (kunskapsöverföring) i tillräcklig mängd för att kunna utföra sitt arbete. Anveckling fungerar tvärtom. Anvecklaren har sin verksamhetskunskap och tillägnar sig utvecklingsverktygskunskap för att kunna utföra sitt arbete. Olika arbetsuppgifter kräver olika kunskap. Olika typer av kunskap ger möjlighet att utföra olika uppgifter. Den som kan "allt" har stora möjligheter. Att lära sig "allt" tar dock tid. Avdic (1995a:119ff) visar att den faktor som mest påverkar kalkylsystemets utformning vad gäller funktion, användningsområde och komplexitet är anvecklarens utvecklingsverktygskunskaper.³⁰

²⁹ Att exemplifiera på detta sätt har för och nackdelar. I bästa fall länkas läsaren in på ett tänkande som förtydligar meningen med begreppen. I sämsta fall låser sig läsaren vid just de uppräknade begreppen och inte vid det generella i uppräknningen. De uppräknade förutsättningarna och effekterna gör inte anspråk på att ge några specificerande associationer. Utan mer att fungera som just anspråkslösa exempel.

³⁰ Förutom anvecklarens verksamhetskunskaper, vilka förutsätts.

Ett rimligt antagande är att man kan relatera de båda kunskapstyperna³¹ till varandra på olika sätt varvid olika kombinationer med olika egenskaper och effekter uppstår. På samma sätt kan det antas att andra yttre förutsättningar kan påverka dessa effekter. Dessa yttre förutsättningar kan t.ex. vara omständigheter som inverkar på den miljö där anvecklaren anvecklar. Exempel skulle kunna vara verksamhetstyp, bransch, IT-mognad, IT-organisation, företagsstorlek, ledningsfilosofi, företagskultur etc.

De specifika forskningsfrågorna lyder:

- **Hur kan anvecklaren med hjälp av kalkylprogramanveckling ta hänsyn till svårformaliserbar kunskap och uppfylla svårformaliserbara mål?**
- **Vilka utvecklingsverktygskunskaper behövs för att utföra olika arbetsuppgifter?**
- **Vilka övriga förutsättningar påverkar anvecklarens möjligheter att utveckla kalkylsystem?**
- **Vilka effekter får kalkylprogramanveckling (KPA)?**

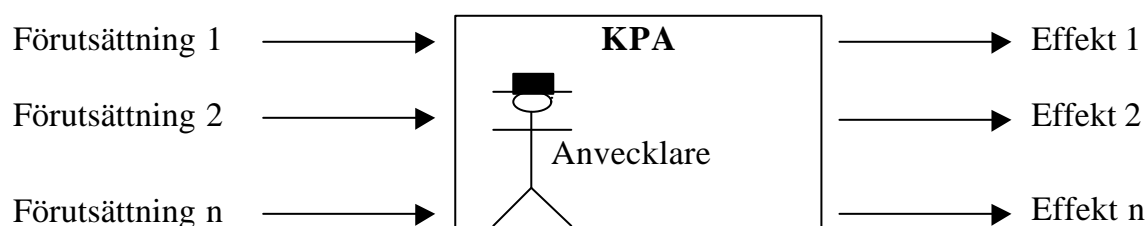
1.4 Syfte

Utgångspunkten för avhandlingen är att fenomenet KPA är en intressant yttring av IT-utvecklingen. Syftet med avhandlingen är att utforska möjligheter och begränsningar med fenomenet. Detta görs i form av påverkande förutsättningar och erhållna effekter. Speciellt uppmärksammas betydelsen av olika kunskapstyper.

Avhandlingens övergripande syfte är att utifrån forskningsfrågorna ovan

formulera en begreppsmodell för vilka förutsättningar, i form av företeelser och förhållanden, som påverkar kalkylprogramanveckling och dess resultat med avseende på möjligheten att utföra arbetsuppgifter. Modellen skulle kunna kallas för en anvecklargenerisk modell.

Den utvecklade kunskapen syftar ytterst till att öka förståelsen för möjligheter och begränsningar med fenomenet kalkylprogramanveckling specifikt och anveckling generellt samt att visa praktiska användningsområden för KPA (se Figur 10 nedan).



Figur 10 Schematisk beskrivning av avhandlingens övergripande syfte.

Dessutom syftar avhandlingen mer specifikt till att (inom ramen för ovanstående övergripande ansats och i relation till frågeställningarna ovan)

- **visa hur hänsyn kan tas till svårformaliserbar kunskap och svårformaliserbara mål**

³¹ Som i sin tur kan kategoriseras på olika sätt.

- **relatera verksamhetskunskap och utvecklingsverktygskunskap i ett kalkylprogramanvecklingssammanhang.**
- **visa praktiska användningsmöjligheter med KPA**

Detta syfte bygger på två teser, nämligen:

1. Det finns möjlighet att ta hänsyn till svårformaliserbar kunskap vid kalkylprogramanveckling.
2. Kunskaper om utvecklingsverktyg skapar nya möjligheter att utföra arbetsuppgifter.

Den utvecklade kunskapen kan utnyttjas till att öka förståelsen för fenomenet kalkylprogramanveckling samt öka kunskaper om möjligheter och begränsningar med kalkylprogramanveckling specifikt och anveckling generellt. Dessutom kan den utvecklade kunskapen användas för att prioritera inläring av specifika typer av utvecklingsverktygskunskap för att kunna bidra till att utföra vissa typer av arbetsuppgifter.

Målgruppen indelas i en primär och en sekundär mängd. Med primär målgrupp menas den mängd personer som förväntas kunna läsa och förstå avhandlingen och samtidigt ha någon nytta av kunskaperna. Med sekundär målgrupp menas den mängd personer som kan tänkas ha nytta av att känna till innehållet i avhandlingen. Den primära målgruppen är forskare, lärare, studenter samt en del praktiker. Avhandlingen är avsedd att bidra till kunskapsbildningen inom informatikområdet och skriven så att den primära målgruppen förväntas förstå innehållet i avhandlingen och ha nytta av det i en framtida forskar- eller yrkesverksamhet. Nyttan i yrkesverksamheten ligger i att känna till dels de möjligheter som anveckling innebär i en verksamhet och dels de bakomliggande fenomen som påverkar förutsättningar för systemutveckling. Den sekundära målgruppen utgörs av anvecklare, verksamhetsansvariga och forskare med inriktning mot informatik, beslutsstöd och organisation.

1.5 Disposition

Avhandlingen består av fyra delar. Del I, *Grunderna*, innehåller detta introduktionskapitel samt ett metodkapitel innehållande en redogörelse för metod som använts för att besvara forskningsfrågorna.

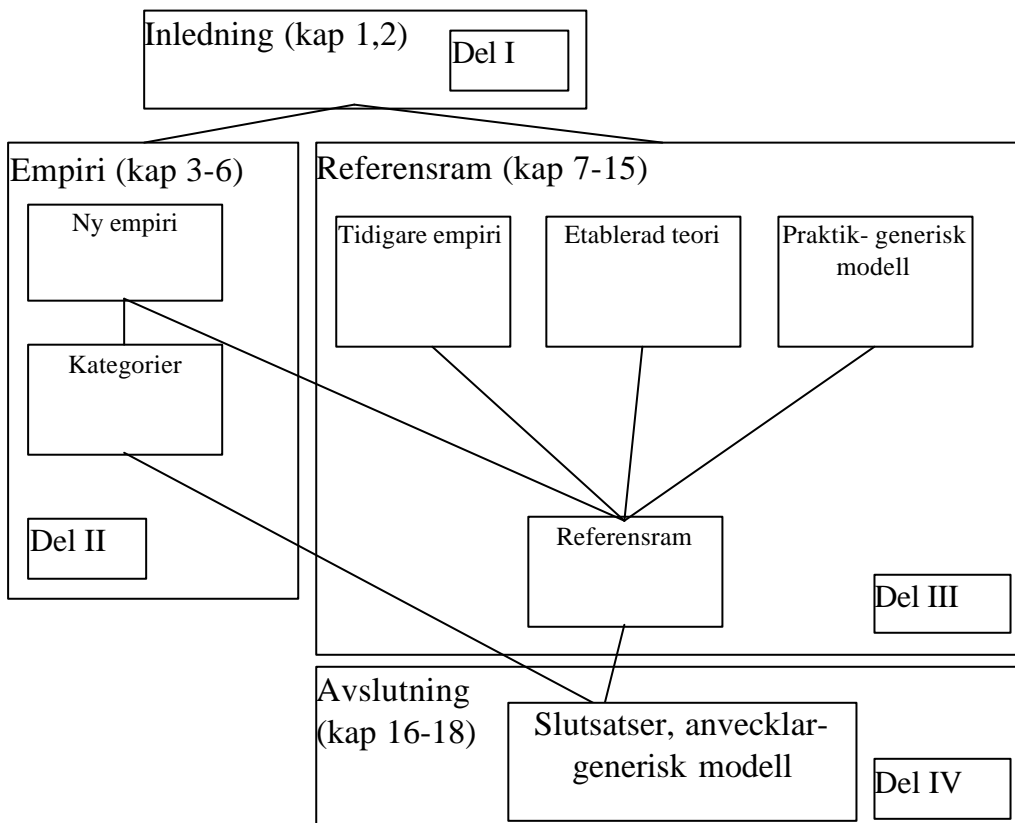
Del II, Empiri, redovisar resultatet av de fyra studier som utförts inom ramen för avhandlingsprojektet. I empiridelen redovisas resultat från datainsamling som har ambitionen att vara Grounded Theory-influerad. Detta innebär att undersökningsuppläggningsen gör anspråk på att vara i viss mån förutsättningslös, vilket diskuteras närmare i kapitel 2. Förutsättningslösheten tar sig uttryck i en eftersträvd öppenhet mot empiri. Inga uttalade modeller har medvetet använts vid datainsamlingen. I empiridelen redovisas resultaten medvetet deskriptivt till viss del. Såväl empirin som de ur empirin genererade kategorierna har styrt diskussionen av en referensram för kalkylprogramanveckling. Av dessa skäl har empiridelen placerats före referensramsdelens.

Del III, Referensram, innehåller nio kapitel där en referensram för kalkylprogramanveckling diskuteras och presenteras. Referensramens utformning bygger på den prak-

tikgeneriska modellen (Goldkuhl & Röstlinger 1998) och kapitelindelningen följer en modifierad variant av denna. Eftersom empirin utnyttjats vid formuleringen av referensramen följer referensramen efter empirikapitlet. Etablerad teori tolkas i denna del utifrån redovisad empiri. Valet av referensmodell och därtill hörande teoriavsnitt har styrts av det empiriskt studerade fenomenet kalkylprogramanveckling. Ytterligare ett skäl till att placera referensramen närmare slutet av avhandlingen är att referensramen som sådan kan ses som en del av avhandlingens resultat. Vid formulandet av referensramen används förutom empiri redovisad i empiridelen även empiri från projektets del ett, som redovisats i min licentiatavhandling (Avdic 1995a). Etablerad relevant teori relateras till KPA i denna del.

I *Del IV, Avslutning*, analyseras den presenterade empirin och teorin i avsikt att uppfylla syfte samt besvara forskningsfrågorna. I denna del sammanfattas även avhandlingen på engelska.

Figur 11 nedan visualiserar sambandet mellan de olika delarna:



Figur 11 Samband mellan avhandlingens delar.

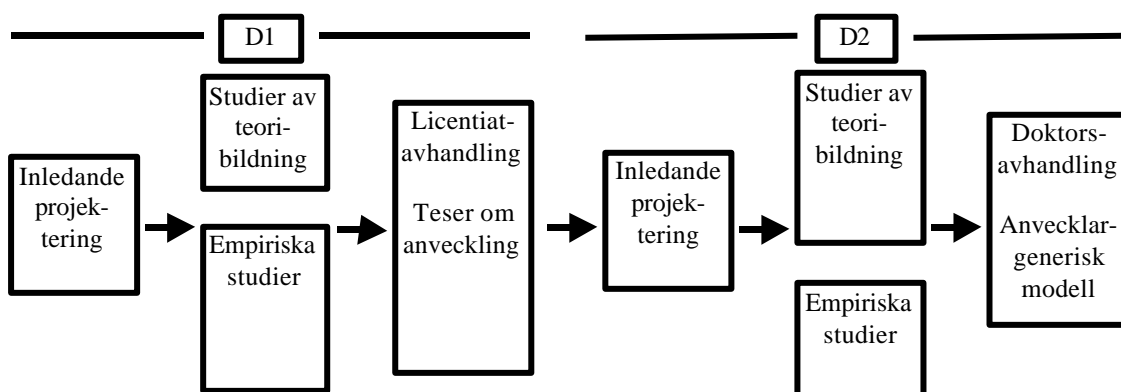
"Avhandlingen kan sägas vara huvudsakligen kvalitativ, hermeneutisk och abduktiv till sin karaktär."
(Avdic 1999:35)

- *Detta kapitel syftar till att beskriva och motivera den forskningsmetod som använts i avhandlingen.*

2 Forskningsmetod

Forskningsarbetet som ligger till grund för denna avhandling kan sägas vara huvudsakligen kvalitativt i en huvudsakligen hermeneutisk metodtradition. Forskningen bedrivs utifrån en abduktiv metodansats, vilket innebär att såväl deduktiva som induktiva inslag förekommer.

2.1 Projekt och delprojekt



Figur 12 Översiktlig arbetsgång för forskningsprojektet Kalkylprogramanveckling.

Avhandlingsarbetet ingår som en del i ett större projekt, Kalkylprogramanveckling, som inleddes med delprojekt 1 (D1), vilket utmynnade i en licentiatavhandling (Avdic 1995a). Projektet avslutas med delprojekt 2 (D2), vilket resulterat i denna doktorsavhandling. Projektet beskrivs översiktligt i Figur 12 ovan. Under D1 har tonvikten legat

på empiriska studier medan teoretiska studier har fokuserats mest under D2. Det inledande delprojektet kan sägas ha haft en explorativ strategi då studieområdet under detta projekt kunde uppfattas som utforskat. Metodasppekter rörande D1 finns beskrivna i Avdic (1995a). D2 behandlas nedan.

2.2 Strategi

En grundläggande fråga i ett forskningsprojekt kan formuleras: 'Givet ett visst syfte eller en specifik frågeställning, vilket är det mest ändamålsenliga sättet att uppnå detta?' Projektets resultat beror till stor del på den valda metoden. Olika metoder kan ge olika resultat.¹ Strategier kan i detta sammanhang ses som en övergripande kategorisering av forskningsmetoder. I ett forskningsprojekt kan val av strategi göras medvetet eller omedvetet. Ett medvetet val av strategi kan medföra en medvetenhet om möjligheter och begränsningar i metodval.

Vid val av forskningsstrategi kan olika perspektiv på valsituationen antas. Valet kan utgå från forskningsfält eller undersökningsmetod (Patel & Tebelius 1987:57ff). Ett annat sätt är att utgå från vilken typ av syfte som gäller genom att analysera vilken karaktär den sökta kunskapen har. Att fastställa detta är inte förenat med någon automatik. Indelningen av kunskapsstyper är i sig ett ställningstagande, som får konsekvenser för valet av metod. Några strategier som förekommer i litteratur om forskningsmetodik (t.ex. Goldkuhl (1998), Lundahl & Skärvad (1982) och Patel & Tebelius (1987)) är

- Beskrivande
- Explorativ
- Förklarande

Nämnda strategier varierar med graden av förkategorisering och är av typkaraktär, dvs de anger huvudsaklig inriktning. Ett visst mått av beskrivning kan ingå i ett forskningsprojekt även om det är huvudsakligen explorativt eller förklarande. En explorativ undersökning skulle kunna ha förklarande inslag etc.

Med en *beskrivande strategi* har inte forskaren en uttalad ambition att förklara eller förstå ett specifikt problemområde utan snarare att beskriva ett på förhand definierat område. Ambitionen att enbart beskriva utesluter formulering av teorier om orsaker och innebörder. "Förklaringar är vad som skiljer en teori från beskrivning." (Wallén 1996:53). Med en beskrivande strategi kan den sökta kunskapen huvudsakligen benämnas som deskriptiv. En vanlig metodansats för deskriptiva strategier är surveyundersökningar (Patel & Tebelius 1987:54). Ett visst mått av beskrivande ingår i de flesta forskningsarbeten.

Med en *explorativ strategi* har forskaren ambitionen att utforska ett mindre utforskat problemområde i avsikt att

¹ Ett klassiskt exempel på detta är försöken att fastställa vad elektroner är. Beroende på mätmetod befanns elektroner vara partiklar respektive vågrörelser.

Forskningsmetod

- formulera hypoteser om området
- generera kunskap om problemområdet
- utforska och sammanställa befintlig kunskap om problemområdet.

Metodtrianglering (se nedan) förknippas med explorativa undersökningar. Metoder som kan komma ifråga är intervjuer, litteraturgenomgångar och fallstudier (Lundahl & Skärvad 1982:60).

Med en *förklarande strategi* har forskaren ambitionen att undersöka samband mellan faktorer. Kraven på hur strikt detta skall tolkas kan variera mellan discipliner. Inom naturvetenskapen finns en stark förklaringsstradition. Man strävar efter att kartlägga orsaker till fenomen exempelvis inom medicinområdet. Inom samhällsvetenskapliga ämnen kan kraven på förklaringsgrad inte alltid ställas lika strikt som inom naturvetenskapen. Man kan då tala om påverkande effekter snarare än orsaker. En variant av förklarande strategi är *hypotestestande strategi* där det gäller att pröva huruvida ett antagande som gäller för problemområdet kan anses överensstämma med empiriska data och eller övergripande teorier (Wallén 1996:71). Om hypotesen inte överensstämmer förkastas den. Viss form av kvantitativ hypotestestning förknippas inte sällan med statistiska undersökningar och experiment.

Avhandlingsarbetet kan å ena sidan inte entydigt placeras in under någon av ovanstående strategier. Å andra sidan kan element av alla tre strategierna sägas ha viss giltighet för avhandlingsprojektet. *Explorativa element* finns i och med att studieområdet kan anses som relativt utforskat och i avsaknad av en etablerad begreppsarsenal. D1 har, som nämnts ovan, genomförts efter en huvudsakligen explorativ strategi men även D2 har haft ambitionen av viss förutsättningslöshet i förhållande till datainsamlingsarbetet. Med förutsättningslöshet avses dock inte total förutsättningslöshet i förhållande till studieområdet. Innehållet i eventuella intervjufrågor är inte slumpvis utvalt och respondenter utses inte genom lottdragning. *Deskriptiva element* finns i och med att beskrivning och redovisning av omständigheter sker, i synnerhet i empiridelen. Deskription har dock inte ett egenvärde. Den som beskriver har normalt bakomliggande skäl till beskrivningar. Beskrivningar görs inte ad hoc. Beskrivningarna i denna avhandling syftar till att ligga till grund för den anvecklargeneriska modell som presenteras i den avslutande delen. Urval av beskrivningar har gjorts för att uppfylla syftet med avhandlingen. Om det inte finns *hypotestestande element* så finns det åtminstone testande element i D2 i och med att det som en utgångspunkt för D2 finns ett antal teser som beskrivs i kapitel 1.3. Att inte utgå från några utgångspunkter eller uppfattningar vore lika orealistiskt som att inte ha syfte med beskrivningar. En utgångspunkt för forskningsarbetet har varit den praktikgeneriska modellen (se kap 7). Skälet till att den praktikgeneriska modellen valts som utgångspunkt diskuteras i samband med att den presenteras. En förklarande metodansats finns i avhandlingen i och med att den sökta begreppsmodellen pekar ut förutsättningar och konsekvenser för KPA.

Avhandlingsarbetet syftar till begreppsbildning inom ett förhållandevis nytt studieområde. Teori och empiri har valts utifrån detta syfte. Den sökta begreppsmodellens syfte är att göra studieområdet begripligt och intellektuellt hanterbart. Intellektuellt hanterbart kan området bli om det relateras till näraliggande begrepp och om begrepps-

dellen innehåller meningsfulla och belysande begrepp. Behovet av en begreppsmodell ökar i och med att studieområdet KPA dels är relativt outforskat och dels saknar en etablerad och allmänt omfattad begreppsarsenal (se kap 10). Ett exempel på detta, är att området kalkylprogramanveckling inte har någon naturlig hemvist i klassifikationskoder på bibliotek, i sökord vid databassökning eller vid Internetsökning. Inte heller finns några konferenser, vetenskapliga tidskrifter eller "special interest groups" som naturligt behandlar det sökta problemområdet. Dessa omständigheter får konsekvenser för metodval. Exempel på sådana konsekvenser är att öppenhet mot empiri och variation i metodanvändning kan ses som motiverad. De val som denna öppenhet och variation resulterat i diskuteras närmare nedan.

Ovanstående diskussion kring avsaknaden av begrepp inom studieområdet kan motivera en introduktion av begreppet *begrepp*. Med begrepp avses delar av människors tankeinhåll, som motsvarar ting, egenskaper eller processer i verkligheten² (Rosing 1978:38ff). För att tala om verkligheten (via begrepp) krävs ett språk bestående av ord. Betydelsen av ett ord är begreppet. Förhållandet mellan term och begrepp kallas semantik (se Tabell 2 nedan). Verkligheten träder fram då begreppen formuleras och ges representation i form av ord. Ett verksamhetsområde är inte intellektuellt hanterbart om det inte kan refereras till med tankekonstruktioner i form av begrepp.

Tabell 2 Begrepp i ett sammanhang (Rosing 1978:39).

Nivå	Helhet	Del
Språk	Satser	Ord, Termer
Tankeinhåll	Påståenden	Begrepp
Verkligheten "Världen"	Sakförhållanden	Ting, Egenskaper, Processer

Formulerandet av en begreppsmodell representeras av ett antal termer som gör begreppsmodellen tillgänglig för analys, bearbetning och kritik. Begreppsmodellen gör på så sätt studieområdet tillgängligt för studier och fördjupad begreppsbildning. En nackdel med en begreppsmodell är att den leder in betraktaren på ett specifikt sätt att se studieområdet och avgränsar därmed i viss utsträckning från andra alternativa synsätt. Om begreppsmodellen belyser ett studieområde som tidigare inte belysts, kan modellen få en måttstocksfunction mot vilken jämförelser sedan kan göras. En förhoppning med avhandlingen är att en grund till begreppsutveckling inom KPA skall läggas.

Litteratur inom området forskningsmetodik organiseras inte sällan kring förekomsten av några inom området välkända motsatspar som berör forskningsideal och metodtradition (Patel & Tebelius 1987, Mårtensson & Nilstun 1988, Svenning 1996, Goldkuhl 1998, Alvesson & Sköldberg 1994, m.fl.). Sådana motsatspar är

- Kvantitativ metod - kvalitativ metod
- Positivism - hermeneutik

² Här antas att det finns en objektiv verklighet som människor förmår att uppfatta på ett likartat sätt.

- Deduktion – induktion

Det senare motsatsparet kompletteras av Alvesson & Sköldbberg (1994:43ff) med begreppet abduktion. Avhandlingen kan sägas vara huvudsakligen kvalitativ, hermeneutisk och abduktiv till sin karaktär. Kategorierna behandlas i de följande avsnitten.

2.3 En kvalitativ metodansats

Forskningsarbetet bakom avhandlingen kan sägas följa en huvudsakligen kvalitativ metodansats. Med en kvalitativ metodansats avses här en metodansats som utmärks av närhet till forskningsobjektet (Holme & Solvang 1997:76) och förståelse av sociala fenomen (Patton 1980:19). Kvalitativa metoder framställs ofta i motsats till kvantitativa, vilka är vanliga inom naturvetenskaplig forskning och ofta innebär statistisk analys. För att detta skall vara möjligt måste insamlad information omvandlas till siffror och mängder. Kritik mot kvantitativ forskning rör ibland den förmenta ambitionen att se vetenskapen "...som en social, opolitisk domare, som klart kan skilja på rätt och fel." (Svenning 1996:28). En kvalitativ undersökning kännetecknas av att den söker fånga icke-kvantitativa aspekter av någon verksamhet. Den utgår normalt inte från hypoteser utan strävar ofta efter att inte förkategorisera studieområdet (Bogdan & Taylor 1972:25ff). Patel & Tebelius (1987:43) beskriver distinktionen mellan kvantitativ och kvalitativ forskning på följande sätt: "I relation till val av kvantitativ inriktning kan man förenklat säga att forskaren antingen söker kunskap som skall mäta, beskriva och förklara fenomen i vår verklighet eller så söker han kunskap som skall inventera, uttyda och förstå fenomen". Valet mellan kvalitativ och kvantitativ metodansats styrs främst av vald frågeställning och valt syfte.

2.3.1 Kvantitativ och kvalitativ

Tabell 3 Kvantitativ och kvalitativ metod (Holme & Solvang 1997:78).

	Kvantitativ metod	Kvalitativ metod
1	Precision: forskaren eftersträvar en maximalt god avspegling av den kvantitativa variationen.	Följsamhet: forskaren eftersträvar bästa möjliga återgivning av den kvalitativa variationen.
2	Ringa information om många undersökningsenheter; går på bredden.	Riklig information om få undersökningsenheter; går på djupet.
3	Systematiska och strukturerade observationer, t.ex. enkät med fasta svarsalternativ.	Osystematiska och ostrukturerade observationer, t.ex. djupintervju eller intervjumall utan fasta frågor eller svarsalternativ.
4	Man intresserar sig för det gemensamma, det genomsnittliga eller det representativa.	Man intresserar sig för det säregna, det unika eller det eventuellt avvikande.

5	Avstånd till det levande: insamlingen av information sker under betingelser som skiljer sig från den verklighet man vill undersöka.	Närhet till det levande: insamlingen av information sker under betingelser som ligger nära den verklighet man vill undersöka.
6	Man intresserar sig för åtskilda variabler.	Man intresserar sig för sammanhang och strukturer.
7	Beskrivning och förklaring.	Beskrivning och förståelse.
8	Åskådare eller manipulatör: forskaren iakttar fenomenet utifrån och strävar efter en roll som observatör. Variationen för variabler kan manipuleras fram.	Deltagare eller aktör: forskaren observerar fenomenet inifrån. Han vet om att han påverkar resultaten genom det faktum att han är närvarande. Han kan även delta som aktör.
9	Jag-det-relation mellan forskaren och den undersökte.	Jag-du-relation mellan forskaren och den undersökte.

I Tabell 3 ovan sammanfattas de av Holme & Solvang (1997:78) identifierade och idealiserade kännetecknen för kvantitativa och kvalitativa undersökningar. Utifrån denna sammanfattning kan följande karakteristik göras av projektet kalkylprogramanveckling:

1. Forskningsprojektet har inte syftat till att kvantifiera kalkylprogramanvecklingarnas verksamhet utan snarare att förstå och tolka vad som händer när människor med stor verksamhetskunskap kan utnyttja detta tillsammans med verktyget kalkylprogram.
2. Med undersökningsenheter kan olika objekt åsyftas. I föreliggande fall är såväl kalkylsystem som anvecklare undersökningsenheter. I D1 var båda typerna av enheter fler än i D2. Undersökningsenheter i D2 i form av kalkylsystem kan inte sägas vara få men syftet har inte uttalat varit att uppnå representativitet. Undersökningsenheterna i D2 i form av respondenter är få, tre per studie i tre av de fyra studierna och något fler i den sista. Informationsmängden är inte ringa.
3. Datainsamling i delprojekt två har skett med djupintervjuer. I det första delprojektet förekom enkäter.
4. Då det inte funnits en ambition att bestämma antalet undersökningsenheter utifrån ett syfte att generalisera kan man inte säga att det finns en generaliserande ambition. Å andra sidan kan det totala antalet kalkylsystem och respondenter i de båda delprojekten sägas vara tillräckligt stort för att inte enbart framstå som exemplifierande. Ambitionen att formulera en modell för att förstå fenomenet kalkylprogramanveckling gör dock anspråk på viss generalitet.
5. De empiriska studierna har syftat till att förstå anvecklarens upplevelse av sin praktik utan förkategoriseringar så långt det går. Metoden för att uppnå detta har varit öppna djupintervjuer där ambitionen varit att i möjligaste mån följa respondentens svarande och övriga agerande³ vid intervjuerna.
6. Intresset har fokuserats mer på sammanhang och strukturer än på åtskilda variabler.

³ Då en stor del av intervjuerna ägt rum vid datorn har respondenten haft möjlighet att visa kalkylsystemexempel på datorn.

7. Här finns visserligen ambitioner att beskriva och förstå, men i strävan att formulera en begreppsmodell innehållande förutsättningar och effekter ligger även en ambition att förklara i termer av påverkande faktorer och effekter.
8. Forskaren har i projektet agerat såväl åskådare som deltagare (det senare dock i mindre omfattning). Det faktum att det funnits inslag av deltagande diskvalificerar projektet från att vara objektivt och värderingsfritt på det sätt som avses i en renodlad kvantitativ studie.
9. Som beskrivs längre fram har relationen mellan forskare och anvecklarna till största delen utmärkts av ett jag-du-förhållande.

Som diskuteras i kapitel 1 är verksamhetsetiska mål och svårformaliserad kunskap i fokus under hela studien. Att endast tolka t.ex. olika kalkylsystem utan att relatera till anveklarens intentioner skulle strida mot syftet med projektet. Kategorierna som avses att utvecklas är formaliserade i den meningen att de syftar till att förstå och strukturera ett område i utveckling och förändring. De teman som behandlas i projektet, t.ex. kunskapsypers roll för möjlighet till anveckling, kan betraktas som förhållandevis komplexa. Dylika teman behandlas inte enkelt med en kvantitativ metodansats.

Alvesson & Sköldberg (1994:12) talar om kvalitativ metod som reflekterande forskning och menar att denna utmärks av de två grundelementen tolkning och reflektion. Elementet *tolkning* innebär att alla referenser till empiri är tolkningsresultat. Detta innebär i sin tur att det inte finns någon entydig relation mellan mätningar, data, intervjupersoners utsagor etc. och något utanför det empiriska materialet. Elementet *reflektion* "...kan definieras som tolkning av tolkning och igångsättande av kritisk självprövning av egna tolkningar (inkl konstruerade) av empiriskt material." (Alvesson & Sköldberg 1994:12). Enligt detta synsätt framstår det som väsentligt att ange från vilka utgångspunkter som tolkning görs så att läsaren skall ha en möjlighet att avgöra relevansen i de presenterade resultaten.

Tolkning har varit ett centralt inslag i projektet då empiri bearbetats (se kap 2.6 nedan). Även de teoretiska inslagen kan sägas bygga på tolkning. Den reflekterande ambitionen har implementerats i form av diskussion av metod och referensram.

2.3.2 Metodtriangulering

Utmärkande för kvalitativa metodansatser är *metodtriangulering* (Wallén 1996:43). Med metodtriangulering avses användande av flera metoder. "Trianguleringen i sin fulla vidd rör data, observationer, teori och metoder." (Svenning 1996:87). Syftet med trianguleringen kan vara att uppnå att slags kontroll av kvalitativa data. Inom kvantitativa metoder är kontroll av data vanligen mindre komplicerad, vilket gör att metodtriangulering inte är lika motiverad eller i varje fall inte lika vanlig inom denna tradition.

Metodanvändning i D1 har varit av såväl kvalitativ som kvantitativ art. Det kvantitativa inslaget har bestått av en enkätundersökning. De kvalitativa inslagen har bestått av

djupintervjuer, studier av kalkylsystem, retroaktiv deltagande observation samt studier av relaterad teori. Metodanvändningen i D1 redovisas närmare i Avdic (1995a).

I D2 har följande metoder använts

- Studier av relaterad teori
- Djupintervjuer
- Studier av kalkylsystem
- Deltagande observation

Djupintervjuer och studier av kalkylsystem har utförts inom ramen för fyra fallstudier enligt Tabell 4 nedan.

Tabell 4 Empiriska studier i projektets del två.

Fallstudie	Beskrivning
Applikation (1)	Djupintervjuer med deltagare i ett anvecklarinitierat utvecklingsprojekt samt deltagande observation
Myndighet (2)	Djupintervjuer med tre anvecklare vid en myndighet och studier av dessas kalkylsystem
Industri (3)	Djupintervjuer med tre anvecklare vid en processindustri och studier av dessas kalkylsystem
IT-specialist (4)	Djupintervjuer med tre IT-specialister

Studierna har genomförts i den ordning de beskrivs i Tabell 4 ovan. Studie tre och fyra har skett parallellt. Empirinära metod beskrivs närmare nedan i kapitel 2.6. De specifika studierna beskrivs närmare i *Del II* i samband med respektive studie.

2.4 En hermeneutisk metodansats

Forskningsarbetet bakom avhandlingen kan sägas ha betydande inslag av en *hermeneutisk metodansats*. Hermeneutik som vetenskapsideal framställs ibland som motsatt till positivism (Mårtensson & Nilstun 1988:37, Hägg, Wiedersheim-Paul & Aronsson 1977:59). Hermeneutik beskrivs inte sällan som växelvis *tolkning* och *förståelse*, som kan ske ömsesidigt i en process, ibland kallad den *hermeneutiska spiralen* (Ödman 1979, Goldkuhl 1990:4ff, Helenius 1990:64ff). Den hermeneutiska spiralen fungerar på så vis att utifrån en viss *förförståelse* fokuseras växelvis helhet och delar så att en tolkning av det observerade leder till ökad förståelse. Förförståelse beskrivs av Goldkuhl (1990):

"Förförståelse består av kategorier och uppfattningar. Kategorier (begrepp) är strukturer att inordna intryck i. Att bilda begrepp innebär en generaliserings- och abstraktionsprocess... Man inriktar sig på de egenskaper som *förenar* olika element i klassen och samtidigt *särskiljer* dem från andra som inte ingår i klassen." (Goldkuhl 1990:7)

Begreppen, som förförståelsen utgörs av, kan sägas utgöra en form eller en modell med vars hjälp vi placerar intryck på plats och relaterar dem till varandra (tolkning).

Ödman (1979:77) jämför med att lägga pussel, där den färdiga bilden av hur pusslet skall bli, kan sägas utgöra modellen. Förförståelsen styr vårt observerande och avgör vilka aspekter som läggs på de studerade fenomenen. Till förförståelsen hör ett annat fenomen, intentionaliteten. Intentionaliteten "...kan definieras som 'den struktur som ger mening åt upplevelsen'. Intentionaliteten handlar om medvetna och omedvetna avsikter, som levande och upplevande varelser kan ha och som påverkar tolkningsprocessen (Ödman 1979, Patel & Tebelius 1987:34).

I den hermeneutiska spiralen sker växelvis tolkning och förståelse utifrån den tolkandes intentionalitet. Mellan tolkning, förståelse och intentionalitet råder ett dialektiskt förhållande (Helenius 1990:76). Ett dialektiskt förhållande kännetecknas av skiftning mellan motsatser enligt mönstret "tes-antites-syntes" (Ljungdahl 1947).

Tolkandet sker genom språket. Språket "...ligger till grund för vår livssyn och vår förståelse av tillvaron och utgör en garanti för att vi tillhör en social gemenskap" (Patel & Tebelius 1986:33). Då tolkning och förståelse sker, sker det utifrån någon språklig konstruktion som upplevs av den tolkande.

"Tolkningens föremål är vanligen inte exakt det som skall tolkas eller det som hänt utan spår av händelser. Mest är det fråga om texter." (Helenius 1990:70)

Helhet och delar är centrala begrepp inom hermeneutiken. Delarna formar helheten och tolkas utifrån en uppfattning av helheten. Det sker ett växelspel mellan delar och helhet i tolkningen som leder till ökad förståelse. Skälet till att man vill fokusera växelvis är att olika perspektiv framhäver olika egenskaper och tolkningsprocessen tillförs insikter som leder till djupare förståelse då perspektiven växlas. Fokuseringen på helhet/del kan liknas vid vad man inom psykologin kallar gestaltväxling (Patel & Tebelius 1987:34). Gestaltväxling innebär att något av två relaterade perspektiv intogs växelvis. Då det ena perspektivet är i förgrunden och är föremål för studier är det andra perspektivet i bakgrunden. I nästa moment växlar så observatören perspektiv och fokuserar på det andra perspektivet. Det är dock viktigt att inse att bakgrundsperspektivet hela tiden finns där precis som på den klassiska bilden 'Min fru och min svärmor' (Lundahl & Skärvad 1982:38), som kan illustrera denna typ av perspektivväxling, som utmärker det hermeneutiska forskningsarbetet. En annan typisk egenskap för hermeneutisk forskning är att det till skillnad från positivistisk forskning medvetet används en subjektiv verklighetsuppfattning. När det gäller det positivistiska forskningsidealet, uppfattas den objektiva, teorilösa observationen som ett ideal (Patel & Tebelius 1987:30f).

2.4.1 Förförståelse

Förförståelsen har, när det gäller forskningsarbetet bakom denna avhandling, utgått från en modell av fenomenet "människor som arbetar med kalkylprogram". Denna modell har skapats dels genom praktiskt arbete med kalkylprogram, dels genom utvecklande av teoretiska begrepp, som beskriver t.ex. systemutveckling och informationssystem. Utvecklandet av dessa begrepp har skett genom praktiska och teoretiska

studier av olika slag⁴ samt genom samvaro med andra människor med erfarenheter av systemutveckling och informationssystem. De mest grundläggande begreppen i detta sammanhang är just *systemutveckling* och *informationssystem*. Andra grundläggande begrepp är *användare*, *utvecklare*, *kalkylprogram*. Begreppen *kalkylsystem*, *arbetsintegrerad systemutveckling med kalkylprogram* (AIS-K) och *kalkylprogramutveckling* (KPA) kan sägas ha utvecklats under forskningsarbetets gång. En mer fördjupad redogörelse för dessa och andra begrepp finns i avhandlingens *Del III*.

Avhandlingens hermeneutiska startpunkt ligger drygt elva år tillbaka i tiden. Vid examinationen av en kurs i standardsystem vid Högskolan i Örebro, redovisade tre studenter PC-kalkylprogrammet Quattro Pro. 1988, det år examinationen ägde rum, var mini- och stordatorer fortfarande dominerande på företag och hos myndigheter. Persondatorer var i allmänhet stand-alone installationer med prestanda och kapacitet på en nivå som gjorde dem till intressanta verktyg, som dock inte på allvar konkurrerade med större datorer. Det som gjorde kalkylprogrammet intressant var möjligheten att göra flexibla beräkningar av s.k. "what-if" typ. Något år senare kom jag i kontakt med en ekonom som på egen hand med hjälp av ett kalkylprogram, tillverkat ett budgeteringssystem för ett företag med ca 300 anställda. Min bakgrund var (vid båda observationstillfällena) systemvetarens, med systemvetarens metodtradition som referensram. Mot bakgrund av denna metodtradition fanns det skäl att reagera inför det faktum att ett informationssystem av den karaktär som budgetsystemet representerade kunde tillverkas av en person utan utbildning inom ADB-området.

Med utgångspunkt från ovanstående händelser har en referensram och en förförståelse vuxit fram kring fenomenet användning av kalkylprogram. Referensramen har kommit att innehålla fler och fler exempel på personer utan specifika IT-kunskaper som konstruerat kalkylsystem av varierande komplexitet. Då jag antogs till forskarutbildning 1991, fann jag att nämnda studieområde kunde fokuseras som ett forskningsområde. Inom ramen för olika kurser utfördes olika typer av studier av mer strukturerad karaktär, än vad som varit fallet innan forskarutbildningen inleddes. Så småningom började systematisk kunskap växa fram och i och med min licentiatavhandling (Avdic 1995a) hade en explorativ grund lagts till studieområdet *Arbetsintegrerad systemutveckling med kalkylprogram*. För att gå vidare med kunskapsutvecklingen inleddes planeringen av doktorsavhandlingen. Efter licentiatavhandlingen fanns behov av att fördjupa såväl de teoretiska som de empiriska delarna av studieområdet, då licentiatavhandlingen endast var att betrakta som en explorativ studie med tonvikt på systematisering av empiri.

En del i förförståelsen av problemområdet är mina egna erfarenheter av användning av kalkylprogram. Jag har under en tioårsperiod kommit i kontakt med kalkylprogram som användare, utvecklare, forskare, undervisare och författare av undervisningsmaterial. Denna verksamhet har gett mig en förförståelse av problemområdet som kan inverka på olika sätt på kunskapsutvecklingsprocessen. Å ena sidan har min verksamhet

⁴ Exempel på teoretiska studier är akademiska kurser, forskarutbildningskurser och läsande av litteratur inom ämnesområdet ADB. Exempel på praktiska studier är utvecklandet och/eller användande av informationssystem av olika slag.

gett mig kontakter med anvecklare och viss insikt i potentialen hos kalkylprogram. Å andra sidan kan mina kontakter med kalkylprogram ha förhindrat mer förutsättningslösa observationer av kalkylprogramanveckling. Mina relativt rikliga erfarenheter kan (ofrivilligt) ha styrt tolkningar och reflexioner kring kalkylprogramanveckling. En tänkbar positiv effekt är att djupintervjuerna, som ligger till grund för empirin i avhandlingen, blivit mer innehållsrika med fler följdfrågor och med rikligare problematisering av hur kalkylprogrammet använts.

En annan förförståelseaspekt som påverkat det empiriska arbetet är att de studier som gjorts inom ramen för del två av projektet, har utförts i organisationer där jag haft vissa verksamhetskunskaper och viss personkännedom. Dessa kunskaper har underlättat förståelsen av behovet av verksamhetskunskap vid kalkylprogramanveckling. Denna aspekt behandlas utförligare nedan i kapitel 2.6.

2.4.2 Perspektivväxling

Utifrån en förförståelse av fenomenet "människor som arbetar med kalkylprogram" har en tolkningsprocess ägt rum, som fokuserat på motsatta men relaterade begrepp av olika slag. På det mest generella planet har tolkningsprocessen omväxlande fokuserat på, å ena sidan *teori/empiri*, å andra sidan på *process/produkt*. Detta kan sägas vara uttryck för en *hur*-aspekt (hur studieområdet studeras, teoretiskt eller praktiskt) och en *vad*-aspekt (vad inom studieområdet som studeras, processen eller produkten). När det gäller dessa två dimensionspar måste hela tiden något av de båda vara i förgrunden. Om man t.ex. behandlar kalkylsystem (produkten i förgrunden) så görs detta teoretiskt (genom att t.ex. studera HIS enligt Nurminen (1988)) eller empiriskt (genom att genomföra en undersökning). Är det å andra sidan så att man studerar hur någon går till väga för att bygga ett kalkylsystem (processen i förgrunden) så sker även detta antingen teoretiskt eller empiriskt, så är produkten närvarande, fast i bakgrunden.

På ett lägre plan har det också skett växelvis fokusering vad gäller helhet och delar. Ett exempel på växelvis fokusering vad gäller produkt på ett empiriskt plan är när man vid ett tillfälle skärskådar en specifik kalkyl byggd av en användare för ett specifikt behov t.ex. uppföljning (del), och när man vid ett annat tillfälle studerar den process genom vilken kalkylsystem kommer till.⁵ Resultatet av dessa studier, tillsammans med andra teoretiska och empiriska resultat, leder till ett klassificerande av t.ex. specifika egenskaper hos kalkylsystem gjorda för uppföljning (helhet). Detta kan i sin tur eventuellt leda till en insikt om att en term för att uttrycka ett begrepp saknas.

Gestaltväxling/växelvis fokuserande på det högre planet kan t.ex. innebära att när det ena perspektivet (t.ex. produkten kalkylsystem) är i förgrunden ena stunden och är föremål för studier (t.ex. kategorisering av olika typer av kalkylsystem) är det andra perspektivet (processen kalkylprogramanveckling) i bakgrunden. I nästa moment växlar så observatören perspektiv och fokuserar på processen (t.ex. genom att studera olika användares arbete med att utveckla kalkylsystem). Det är lika omöjligt att betrakta ut-

⁵ Dessa erfarenheter existerar i form av skriftligt nedtecknade intervjuer. Analysen sker alltså genom analys av språkliga begrepp, formaliserade i en text.

veckling av kalkylsystem och samtidigt utesluta själva kalkylsystemet ur sammanhanget som det motsatta, att betrakta kalkylsystemet och utesluta processen varigenom det konstruerats. Den sistnämnda aspekten blir särskilt uppenbar då man beaktar att process och produkt är mer intimt förknippade då det gäller arbetsintegrerad systemutveckling med kalkylprogram än då det gäller traditionell systemutveckling⁶. Inom traditionell systemutveckling sker ofta utveckling under en speciell period av en speciell yrkesgrupp. Då systemet är klart sker vid traditionell systemutveckling *ett överlämnande*, vilket i någon mening separerar process från produkt. Då det gäller KPA är process och produkt integrerade vad gäller utveckling/användning och tidsperiod i hög utsträckning (se även sid 189). Integrationen innebär alltså att det sker ett utvecklande och ett användande samtidigt. Användandet, som sker under den stegvisa förfining som präglar anveckling, innebär att produkterna som används under olika delar av anvecklingsprocessen inte är identiska, vilket gör integrationen av process och produkt mer uppenbar. Produktperspektivet kan följaktligen sägas representera ett statiskt perspektiv och processperspektivet ett dynamiskt. Under processens gång kommer produkten att anta olika former. Därför kan ett studerat kalkylsystem ses som en ögonblicksbild av processens framskridande (se Figur 13 nedan).



Figur 13 Process och produkt: statisk och dynamisk dimension av samma fenomen.

Perspektivväxlingen vad gäller teori/empiri har skett på följande vis. Empiriperspektivet har varit i förgrunden vid utveckling och användning av kalkylsystem samt vid studier av användning av kalkylprogram. Teoriperspektivet har fokuserats då teorier om informationssystem har studerats, dels i samband med forskarutbildning dels i

Tabell 5 Avhandlingens indelningsgrunder.

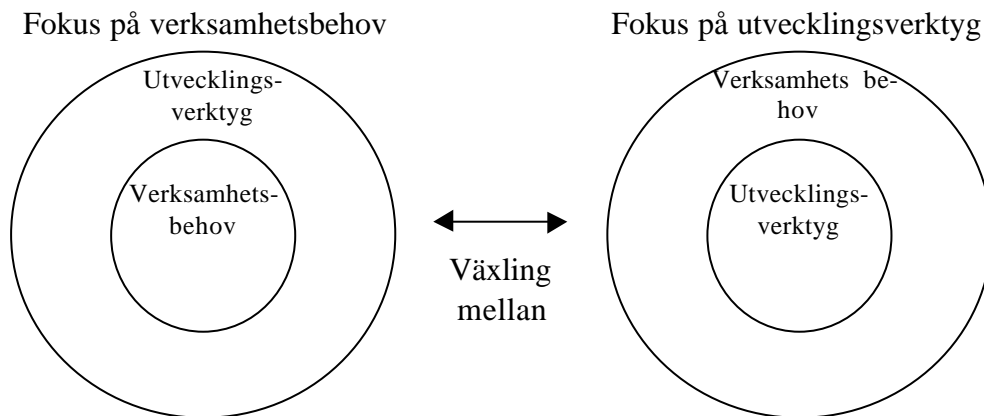
	Empiri	Teori
Process	Redogörelse för resultat av undersökningar om användning av kalkylprogram i syfte att belysa KPA. Se <i>Del II</i> .	Beskrivning av teoretiska aspekter av SU-begreppet för att belysa KPA. Se kap 9 och 10.
Produkt	Beskrivningar av verkliga kalkylsystem för att belysa kalkylsystems egenskaper. Se <i>Del II</i> .	Beskrivning av teoretiska aspekter av IS-begreppet för att belysa kalkylsystem. Se främst kap 8.

samband med formulerandet av referensramen till avhandlingen. Exempel på detta är då Nurminens teorier om det humanistiska perspektivet på informationssystem (Nur-

⁶ Med traditionell systemutveckling avses systemutveckling som utförs av specialister för användare. Traditionell systemutveckling sker normalt efter en kravspecifikation som utformats i en analysfas. Konstruktionen av systemet utförs av programmerare som har programmering till yrke. Se vidare kapitel 9.

minen 1988) studerats och relaterats. Tabell 5 ovan visar avhandlingens indelningsgrunder och hur dessa är kopplade till olika delar av avhandlingen.

Ytterligare en central perspektivväxling, som förekommer i avhandlingen är mellan anveckling ur ett verksamhetsbehovsperspektiv och anveckling ur ett verktygskunskapsperspektiv. Ur ett *verksamhetsbehovsperspektiv* betraktas anveckling och kalkylsystem utifrån uppfattade behov i verksamheten. Teoretiska och empiriska studier ur detta perspektiv observerar om och hur behov kan tillgodoses med kalkylprogramanveckling och kalkylsystem. Detta är ett vanligt synsätt för traditionell systemutveckling. Ur ett *verktygskunskapsperspektiv* betraktas t.ex. kalkylprogram och anveckling som en möjliggörare. Studier ur detta perspektiv fokuserar i första hand verktygs potential för att se huruvida möjligheter erbjuds att tillgodose befintliga eller potentiella verksamhetsbehov. En visualisering av detta förhållande kan ses i Figur 14 nedan. Figuren visar att då verksamhetsbehov fokuseras (den vänstra delen av figuren) är utvecklingsverktyg i bakgrunden, vilket innebär att utvecklingsverktyg, t.ex. kalkylprogram används för att tillgodose verksamhetsbehov. Då utvecklingsverktyg fokuseras (den högra delen av figuren) är verksamhetsbehoven, om inte helt i bakgrunden, så i alla fall inte styrande. I stället övervägs huruvida utvecklingsverktyget innehåller möjligheter som kan komma verksamheten till nytta. Detta kan även benämnas *IT som möjliggörare*.



Figur 14 Olika fokus på verksamhetsbehov och utvecklingsverktyg.

2.4.3 Intentionell aspekt

Den intentionella aspekten av det hermeneutiska forskningsarbetet kan forskaren vara medveten och/eller omedveten om. Den omedvetna aspekten kan vara minst sagt svårt att redogöra för, medan den medvetna har att göra med de mer eller mindre medvetna motiv som författaren har att studera ämnesområdet, i detta fall kalkylprogramanveckling. Nedan görs ett försök att synliggöra de medvetna motiven (intentionerna) bakom avhandlingsarbetet.

Motiven kan sägas ligga på flera plan. Det kan på en nivå gälla val av problemområde. På en annan nivå kan det gälla behandling av problemområde. På en tredje kan det gälla kunskapsbildningsprocessen som sådan.

Då det gäller *val av problemområdet* kalkylprogramanveckling finns en intention att förbättra och underlätta datoranvändning för icke-IT-specialister. Användning av datorer kan ses ur ett samhälleligt perspektiv där den som har kunskaper om IT har fördelar framför den som inte har det. Fördelarna med att ha goda IT-kunskaper kan få effekter på ekonomi, social status, karriärmöjligheter, självuppfattning, barnens möjligheter till utbildning etc. Genom att stödja anvecklare i syfte att söka bidra till att ge dem möjligheter att självständigt, oberoende av IT-specialister kunna utforma informationssystem för att samla in, lagra, bearbeta, distribuera och presentera information kan det ge positiva effekter för personerna ifråga. Då datoranvändning ökar i samhället finns ett behov av att utveckla kunskap om hur icke-specialister kan utnyttja IT.

Då det gäller *behandling av problemområdet* som sådant, finns en intention om att fördjupa förståelsen av ett för mig sedan tio år välkänt område. Det finns en önskan att utnyttja vetenskapliga metoder för att analysera och utveckla kunskap om ett samhälleligt fenomen, som jag uppfattar som intressant av ovan nämnda skäl. Denna önskan utgår från en tro att vetenskaplig metodik kan bidra till att utveckla kunskap och blottlägga sådant som annars skulle vara dolt eller implicit. Avhandlingen utgår från en tro att användning av kalkylprogram är en komplex verksamhet som innehåller strukturer och förhållanden som om de uppmärksammas, benämns och relateras kan bidra till förståelsen av ett, i vårt samhälle, väsentligt område.

Ytterligare ett motiv kan sägas vara en önskan om en djupare förståelse av *kunskapsbildningsprocessen* generellt. Genom forskarutbildning och tillämpning av praktiska och teoretiska kunskaper är målet att tillägna sig generella kunskaper om kunskapsutveckling. Dessa kunskaper kan utnyttjas till vidare forskning om kalkylprogramanveckling eller något annat problemområde. Dessa kunskaper om kunskapsutveckling kan också användas direkt och indirekt i undervisning vid högskola/universitet. Kännedom om kunskapsutveckling har följaktligen också ett slags egenvärde.

Då det gäller avhandlingsarbetet och de projekt som ligger till grund för detta finns en intention att tillämpa etablerad och relevant kunskap för att dels producera giltig kunskap men också för att meritera mig själv som forskare. Ytterligare en aspekt av detta är en önskan att på sikt bidra till inte bara kunskapsuppbyggnad kring kalkylprogramanvändning på icke-IT-specialisters villkor utan även att indirekt bidra till utvecklandet av en systematisk forskningsverksamhet i anslutning till icke-IT-specialisters användning av IT-verktyg. Det finns även en strävan att som forskare behålla den eventuella originalitet som jag besitter i egenskap av min person och min referensram. Det bör, som jag ser det, för läsaren vara möjligt att hitta en person bakom avhandlingens formuleringar. Beslut om forskningens och avhandlingens design skall grundas i egna personliga rationella överväganden och inte vara en frukt av mekaniskt tillämpade kunskaper, även om dessa härstammar från betydelsefulla och inflytelserika forskare. En strävan efter integritet har på detta sätt påverkat utformningen av avhandlingen.

Avhandlingen är vidare ett uttryck för en önskan att utforma avhandlingen som ett uttryck för egen övertygelse om form och innehåll och ett uttryck för mina erfarenheter.

Tillämpningen av forskningsmetodik avser att underordnas mina föreställningar om målgrupp och utformning. För att åstadkomma en sådan form bryts minst två krafter mot varandra. Den ena kraften berör *noggrannhet* och tar sig uttryck i precision, validitet och reliabilitet. Den andra kraften berör *kommunicerbarhet* och berör presentation och läsbarhet. Målsättningen med avhandlingen är att den i sin helhet skall vara tillgänglig för informatikstudenter på C-nivå, men att stora delar av den också skall vara tillgängliga för de personer som avhandlingen handlar om, de anvecklare som bidragit med empiri. Det senare önskemålet är också ett argument för att skriva avhandlingen på svenska i stället för på engelska. Även om avhandlingen skulle kunna tänkas ha lika stor relevans för en icke-svensk publik, skulle jag betrakta det som olämpligt, och oförenligt med mina intentioner, att inte göra avhandlingen tillgänglig för personer, utan vars hjälp avhandlingen inte skulle ha kunnat skrivas.

2.4.4 Subjektiv aspekt

Den subjektiva aspekten av det hermeneutiska forskningsarbetet är relaterad till den intentionella. En hermeneutisk forskare strävar efter att ta fram och synliggöra de personliga erfarenheter och inställningar, som han har och utnyttja dem i forskningsarbetet.

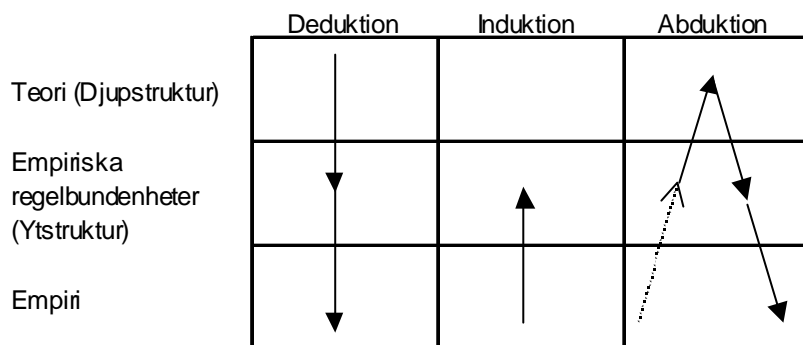
I avsnittet ovan nämndes att ett av motiven till avhandlingen var att stödja icke-IT-specialisters oberoende av IT-specialister. Erfarenheter som kan ha inverkat på denna uppfattning är min bakgrund som sjöman och verkstadsarbetare. Denna bakgrund har gett mig en inblick i levnadsvillkor, som gäller för grupper i samhället som inte har tillgång till möjligheter av olika slag, t.ex. vad gäller boende, att ge sina barn utbildning, arbetssituation och kunskaper om IT. Min subjektiva uppfattning är att det på företag och myndigheter 1999 finns en låg datamognad. Uppmärksammandet av icke-IT-specialisters möjlighet att utnyttja IT på ett självständigt sätt är till viss del ett uttryck för ett ställningstagande för icke-IT-specialisters oberoende.

Vad gäller studierna bakom denna avhandling har studieobjekten valts medvetet utifrån författarens subjektiva uppfattning om vad som kan vara intressanta studieobjekt. Värderingar om personer och företag har använts medvetet för att tolka och skapa möjligheter att tolka och förstå tillstånd och förlopp. En annan subjektiv aspekt är att författaren själv agerat användare, utvecklare och utbildare inom ämnesområdet under en tioårsperiod. Den forskningsmässiga betydelsen av dessa subjektiva erfarenheter berörs i kapitel 2.4.1 och 2.6.

2.5 En abduktiv metodansats

Två vanligt förekommande begrepp vid diskussion om hur kunskap bildas är induktion och deduktion. Med *induktion* avses en process som utgår från att forskaren gör ett antal observationer och utifrån dessa sedan drar generella slutsatser, som kan användas för att tolka "verkligheten". Med *deduktion* avses en process där forskaren utgår från någon typ av teori eller lag, som relateras till empiriska observationer och uttalar sig om den "verklighet" där observationerna görs.

Såväl den induktiva som den deduktiva metodansatsen innebär problem. För den induktiva metodansatsen gäller att hur många observationer forskaren än gör, kan han aldrig vara säker på att nästa observation styrker hans slutsats. Dessutom förutsätter den induktiva processen ett slags teorilöshet vid observationstillfället. Annars skulle det ju vara fråga om deduktion! Att göra helt teoriberoende observationer kan dock knappast anses möjligt, vilket diskuteras nedan. Beträffande den deduktiva metodansatsen kan det anföras att de premisser utifrån vilka slutsatser dras knappast kan ha uppstått av sig själva i tomma luften. Begreppen induktion och deduktion får, för att vara meningsfulla, inte betraktas för strikt utan snarast som en slags övergripande klassificering. En undersökning kan ha övervägande induktiva eller deduktiva inslag (se Figur 15 nedan). En benämning som kan appliceras på en kunskapsbildningsprocess med både induktiva och deduktiva inslag är *abduktion*. Wallén beskriver abduktion som en situation där man står inför en effekt och söker orsaksfaktorer utan att direkt kunna manipulera dessa (Wallén 1996:48).



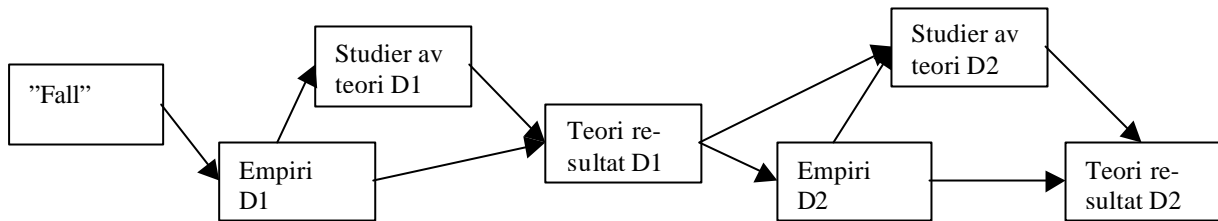
Figur 15 Deduktion, induktion och abduktion (Alvesson & Sköldberg 1994:45).

Enligt Alvesson & Sköldberg (1994) kan abduktion vara den metod som är vanligast vid fallstudier. Abduktion innebär

”...ett (ofta överraskande) enskilt fall tolkas med ett hypotetiskt övergripande mönster, som, om det vore riktigt, förklarar fallet i fråga. Tolkningen bör sedan styrkas genom nya iakttagelser (nya fall). Metoden blir härigenom ett slags kombination av de två tidigare nämnda induktiva och deduktiva, men tillför också nya moment. Under processens gång utvecklas dels det empiriska tillämpningsområdet successivt, dels justeras och förfinas även teorin (dvs det föreslagna övergripande mönstret). Genom inriktningen på underliggande mönster skiljer sig abduktionen fördelaktigt från de båda andra förklaringsmodellerna. Skillnaden är annorlunda formulerat, att den inbegriper *förståelse*.” (Alvesson & Sköldberg 1994:42)

Det abduktiva i föreliggande avhandling ligger i att forskningsarbetet initierats av sk ”fall” (bl.a. en ekonom som byggde ett budgeteringssystem och studenter som redovisade uppgift med kalkylprogram), vilka relaterats till en teoretisk referensram (en systemvetenskaplig sådan). Utifrån denna referensram har ytterligare ”fall” observerats, vilket genererat ytterligare teori (licentiatavhandling). Den utvecklade teorin i D1 har sedan legat till grund för fördjupade empiriska studier i D2. Studierna i D1 har i högre

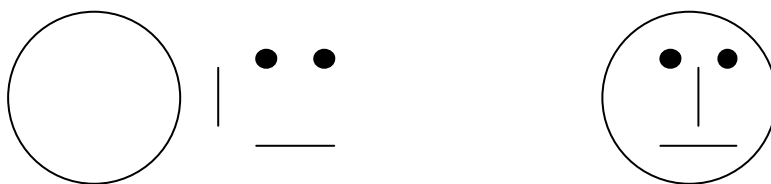
utsträckning haft deduktiva inslag i och med att en surveyundersökning genomförts och genom att fler verksamheter studerats, fast mer ytligt än i D2. Mot bakgrund dels av i D1 utvecklade teori och dels av i D2 utvecklade empiriska resultat har en mer systematisk genomgång av teori utförts i D2, i avsikt att skapa en generisk modell för kalkylprogramanveckling. Denna modell skall sedan kunna användas för att förstå och utveckla miljöer med kalkylprogramanveckling. Den tillämpade kunskapsutvecklingsstrategin beskrivs i Figur 16 nedan.



Figur 16 Schematisk beskrivning av kunskapsutvecklingsstrategi.

I Figur 16 åskådliggörs den empiriorienterade ansatsen av den inbyggda tidsaxeln där avsikten är att demonstrera hur empiriska studier föregår teoretiska. Detta sker dock inte in absurdum utan är snarast uttryck för ett principiellt förhållningssätt. Till problemområdet relaterad teori belyses via empiriska resultat.

Frågan om det är möjligt att göra rena empiriska observationer utan inflytande från uppfattningar eller teorier eller om det är så att data är teoriinfluerade och inte objektiva och fristående från forskaren har diskuterats flitigt i vetenskapsteoretisk litteratur. Ett förekommande sätt att argumentera för att observationer är teoriberoende är att visa fixeringsbilder, där olika betraktare ser olika saker beroende på att de har olika referensram eller där samma betraktare ser olika saker beroende på vad denna väljer att fokusera på bilden⁷. Ytterligare ett exempel på teoriberoende i observationer är tillgång till/kunskap om det sammanhang som observationen görs i. I Figur 17 nedan visas ett exempel som illustrerar detta.



Figur 17 Sammanhang och helhet påverkar tolkning av observationer (Patel & Tebelius 1987:33).

Att hävda att studier av anvecklare som använder kalkylprogram skulle vara oberoende av observatörens referensram förefaller inte särskilt trovärdigt. Det skeende där anvecklaren utför sina arbetsuppgifter och ibland utnyttjar kalkylprogram kan betraktas på ett i princip obegränsat antal olika sätt. Följaktligen kan det fastställas att de observationer som gjorts inom ramen för detta avhandlingsarbete till stor del är en följd av forskarens referensram. Om data är influerade av en referensram i form av värderingar, teorier och uppfattningar så är det, för både forskare och kunskapsmottagare,

⁷ Wittgensteins sk "ankhare" är ett exempel på en sådan fixeringsbild, där bilden kan tolkas som en anka eller en hare beroende på hur betraktaren väljer att fokusera.

intressant att känna till vilka dessa teorier är, så att resultatet kan bedömas med denna vetenskap i beaktande. I avhandlingens *Del III* finns en referensram för studier av kalkylprogramanveckling utvecklad och redovisad. Uppläggningsen av denna teoripresentation styrs i sin tur av den praktikgeneriska modellen (se kap 7). Att just denna modell kommit att utnyttjas är till stor del en följd av empiriska studier. Studier av relaterad teori (som beskrivs i *Del III*) har även haft viss betydelse för det empiriska arbetet. När studier av relaterad teori har varvats med datainsamling har dessa två aktiviteter kommit att växelvis påverka varandra. Ett exempel på detta är kopplingen till olika kunskapsstyper. Studier av teorier om kunskapsbildning (t.ex. Molander 1996) har påverkat tolkningen av de intervjuer som gjorts med utvecklare, vilket i sin tur påverkat tolkningen av teorier om kunskapsbildning.

2.6 Empiri datainsamling

Den kvalitativa metodansats som huvudsakligen inspirerat avhandlingens empiriska arbete är *Teorigenerering på empirisk grund*, som den presenteras i Starrin m.fl. (1991)⁸. Denna teori ansats härstammar ursprungligen från sociologin och har formulerats med avsikt att formulera teorier om sociala processer. Syftet med teorigenerering på empirisk grund är att finna och formulera en teori kring en huvudkategori, som skall vara central för arbetet. Huvudkategorin skall naturligt kunna relateras till övriga funna kategorier. För att finna huvudkategorin som teorin skall byggas upp kring skall insamlade data kodas. *Kodning* innebär att man, genom insamling av data eller bearbetning av text från datainsamlingen, söker hitta centrala begrepp kring vilka teorin kan byggas upp. Kodningen kallas för *öppen* innan man funnit kategorier. Då man börjar koda utifrån vissa specifika koder och tolkar data utifrån dessa kallas kodningen för *selektiv*. Man söker först efter *substantiva* och sedan *teoretiska* koder. Substantiva koder är de centrala begrepp som man inledningsvis finner vid insamling av data eller vid bearbetning av insamlade data. Teoretiska koder är begrepp som relaterar de substantiva koderna till varann. "Teoretiska koder ger omfång, nya vyer och nya perspektiv" (Starrin m.fl. 1991:43). Följande moment kan alltså anses centrala inom nämnda ansats:

- Datainsamling
- Öppen kodning
- Selektiv kodning.

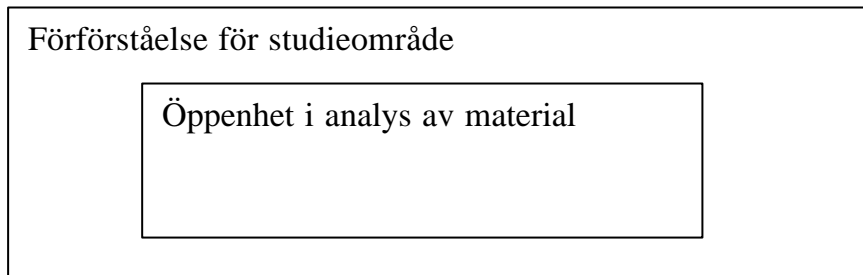
Målet är ett kunna formulera en teori med vars hjälp empiriska data kan förklaras. Teorin och dess variabler skall naturligt passa insamlade data.

"By fit we meant that the categories of the theory must fit the data. Data should not be forced or selected to fit pre-conceived or pre-existent categories or discarded in favor of keeping an existent theory intact." (Glaser 1978:4)

Att teorigenerering på empirisk grund har inspirerat datainsamlingsarbetet innebär inte att alla förutsättningar som relateras till denna metodansats är uppfyllda. I själva verket kan en hermeneutisk och en empiriska metodansats synas motsatta i och med att en

⁸ Som i sin tur bygger på Glaser & Strauss "The discovery of grounded theory: Strategies for qualitative research" (1967).

hermeneutisk metodansats innefattar explicitgörande av förförståelse medan en empirinära metodansats innefattar strävan efter förutsättningslöshet. En ambition att tillämpa det empirinära inslaget har funnits i anslutning till insamling och analys av data från anvecklares erfarenheter. Speciellt centralt har genomgången av kalkylsystem och textmaterial från intervjuer, varit. Genomgången av detta material har utförts enligt ovan nämnda kodningsprinciper, vilket även har resulterat i begrepp och uppfattningar som annars knappast skulle framkommit. Förhållandet illustreras i Figur 18 nedan. Nedan redovisas på vilket sätt i övrigt, som en empirinära metodansats influerat datainsamlingsarbetet.



Figur 18 Förförståelse och öppenhet.

2.6.1 Teoretisk känslighet

En viktig förutsättning vid teorigenerering med empirinära metodik är teoretisk känslighet. "En förutsättning för att uppnå en sådan känslighet är att man börjar varje undersökning med så få förutfattade meningar som möjligt. Det är med andra ord viktigt att vara öppen." (Starrin m.fl. 1991:34). De empiriska studierna som ingår i avhandlingsprojektet kan inte sägas vara utförda utan förutfattade meningar. De två allmänna teser som formulerats i kapitel 1 (se sid 29) har förvisso påverkat undersökningarna. Dessutom kan ett totalt förutsättningslöst närmande till ett problemområde uppfattas som orealistiskt. Vad som är förutsättningslöst beror dessutom på hur problemområdet definieras. I själva definitionen av problemområdet ligger begränsningar av det förutsättningslösa invävda. Ett annat slags öppenhet ligger i själva metodtillämpningen. I detta forskningsprojekt finns en strävan att designa en metod som lämpar sig för det syfte som finns formulerat. Denna metod utgår från en öppenhet i datainsamlingen utifrån från de inledningsvis redovisade utgångspunkterna (se kap 1.3.1) och teserna (se sid 29). En genuin undran över relationer mellan användning av kalkylprogram och verksamhetskunskap har styrt valet och tillämpningen av datainsamlingsmetoder.

Starrin rekommenderar följande tre principer för att öka den teoretiska känsligheten i datainsamlingsarbetet:

1. *Multimetodprincipen* innebär att man använder olika metoder för datainsamling.
2. *Multisinnessprincipen* innebär att man skall använda så många sinnen som möjligt i datainsamlingsfasen.
3. *Principen om estetisk distans* innebär att betraktaren varken har för stor eller för liten distans till det han observerar. (Starrin m.fl. 1991:34)

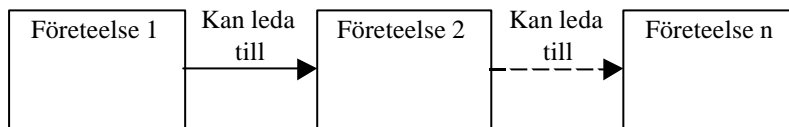
De olika principerna har tillämpats i olika stor utsträckning.

1. Någon strävan att begränsa antalet metoder eller antalet sinnen har inte utmärkt datainsamlingsarbetet. Snarare har medvetenheten om att öppenhet mot empiri, vad gäller datainsamling och bearbetning ökar möjligheten till ett värdefullt resultat av undersökningen, präglad arbetet. Datainsamlingsmetoder som tillämpats under D2 är intervjuer, studier av kalkylsystem, deltagande observation och teoretiska studier (se vidare nedan och kap 2.3.2).
2. Utöver att flera metoder innebär engagemang av olika sinnen så har bearbetningen av empiri inneburit kodning av bandinspelningar och registrering av kalkylsystem, vilket har engagerat flera sinnen. Det faktum att jag träffat flertalet respondenter flera gånger och på så vis blivit bekant med dem, har inneburit att reflektioner kring såväl forskning som KPA kunnat utbytas under olika omständigheter, t.ex. på kafferaster, över något konkret problem med ett kalkylsystem, under en lunch eller då man träffats i affären eller på stan.
3. Då det gäller principen om estetisk distans har närhet till studieobjektens referensram mer utmärkt datainsamlingsarbetet än distans, vilket nämnts ovan och vilket diskuteras nedan. Å andra sidan har inte närheten haft karaktären av deltagande annat än vad gäller projektet Applikation. Men bedömningen har varit att blandning av datainsamlingsmetoder (multimetodprincipen) ökar kvaliteten på resultatet även om det gäller deltagande observation.

Empiriska data har huvudsakligen samlats in på två sätt, via intervjuer och via studier av kalkylsystem.⁹ Insamlingen av empiriska data har skett i två etapper, dels under projektets del ett, som redovisats i en licentiatavhandling (Avdic 1995a) och dels under projektets del två. Diskussioner av metodaspekter för projektets del ett redovisas i Avdic (1995a). Här berörs endast övergripande metodaspekter för projektets del två. Mer detaljerade redogörelser för respektive delprojekt presenteras i samband med resultatpresentationen i *Del II*.

Under arbetet med avhandlingen har teoretiska och empiriska studier varvats kontinuerligt. När det gäller de empiriska studierna har datainsamling och kodning varvats i avsikt att finna teoretiska koder som så småningom lett fram till den huvudvariabel som framstår som den som kan beskriva och förklara fenomenet kalkylprogramutveckling (KPA). Ett sätt att generera koder har varit att visualisera fenomenen konceptuellt genom att göra begreppsgrafer. Denna visualisering har underlättat en hermeneutisk fokusväxling mellan begrepp och förekomst. Genom att konstruera begreppsgrafer enligt mönstret i Figur 19 nedan har analysen kunnat formaliseras och bearbetas ömsom på ett konceptuellt plan och ömsom på ett förekomstplan. Pilen i begreppsmodellerna betyder *kan leda till* om inget annat anges. Genom att dessutom växla mellan undersökningar, mellan personer i undersökningar har begreppsmodeller vuxit fram. Då en begreppsmodell visat brister, t.ex. svag koppling till verksamhetskunskap, har kompletterande intervjuer utförts. Då nya versioner av analysen av anvecklingsverksamheten tillställts respondenterna, har reaktioner från dessa lett till nya insikter om problemområdet.

⁹ Inom ramen för delprojektet *Applikation* insamlades data också med deltagande observation.



Figur 19 Mönster för begreppsmodeller.

2.6.2 Användning av ordbehandlare

En annan metoduspekt vad gäller bearbetning och kodning av data (såväl teoretiska som empiriska) är *användning av ordbehandlare*. Databearbetningen har skett med hjälp av ett ordbehandlingsprogram som tillåtit manipulering av text och grafik på olika sätt. En utgångspunkt för användningen av ordbehandlingsprogram i databearbetningen är att ordbehandling tillför ett kvalitativt element i skrivarbetet jämfört med manuell linjär texthantering (Avdic 1997c). Användningen av ordbehandlingsprogram gör det möjligt att vara icke-linjär i sitt skrivande, vilket kan tilltala personer som är icke-linjära i sitt tänkande. Att medvetet utnyttja det icke-linjära arbetssättet kan uppfattas som en aspekt av databearbetning. Några av de tumregler som utmärkt arbetet med denna avhandling, både i sin helhet och vad gäller empiribearbetning är följande: (Avdic 1997c:1f)

- Handskriv inte kladd. All text skrivs direkt med ordbehandlare.
- Fokusera först på innehåll. Strukturering och formatering kan ske när texten finns nedskrivnen och bearbetningsbar.
- Pröva olika strukturer. Möjligheten att simulera utformning av texten skapar ett större beslutsunderlag för design av den slutliga texten.
- Använd kommentarfunktionen. Idéer skall noteras omedelbart och i anslutning till den relevanta delen av texten men behöver inte tynga den synliga brödtexten.
- Använd rubrikmallar och dispositionsfunktionen. Detta medför flera effekter som kan underlätta bearbetning av text och dess innehåll.
 - A) strukturering av text utifrån innehåll underlättas
 - B) omstrukturering av text underlättas
 - C) skapande av innehållsförteckning och korsreferenser underlättas
- Utnyttja granskningsfunktionerna. För att kontrollera förekomsten av kategoriska uttryck kan sök funktionen användas för att hitta alla förekomster av 'alla', 'inga' etc.
- Utnyttja andra funktioner som index, beskrivningar, korsreferenser, versionshantering etc.

Nackdelar kan också vara förknippade med ordbehandling. Ett frekvent omstrukturerande av en text kan leda till grammatiska fel, syftningsfel och andra typer av felaktigheter. För att undvika detta har texten korrekturlästs regelbundet. I övrigt har säkerhetsåtgärder som är förknippade med informationssystem, t.ex. säkerhetskopiering, gällt även för detta ordbehandlingsdokument.

2.6.3 Huvudvariabel

Under D1 genererades en huvudvariabel, nämligen *integration*. KPA utmärks av integration på flera plan (se sid 190). Denna variabel inverkar på ett flertal effekter av KPA. Kalkylprogramanveckling innebär integration av:

- Verksamhetsansvar, domänkunskap och utvecklingskompetens
- Systemutvecklingsaktiviteter
- Aktörsfunktioner
- Behandlingsfunktioner

Utöver detta genererades att antal kategorier som tillsammans bildade ett embryo till en teori om kalkylprogramanveckling. Resultatet av D1 har utgjort indata till D2, vars resultat redovisas i denna avhandling. Metodmomenten redovisas enligt nedan:

- Beskrivning av datainsamling för studierna finns i anslutning till respektive studie.
- Det stegvisa sökandet efter substantiva och teoretiska koder beskrivs i anslutning till respektive studie. Varje studie avslutas med en sammanfattning. I avhandlingens avslutningsdel sammanfattas den utvecklargeneriska modellen.

2.6.4 Relation till respondent, närhet eller distans?

Då data insamlas genom intervjuer kan olika förhållningssätt till respondenten antas. En indelning är i termer av *närhet* eller *distans*. Enligt en positivistisk metodsyn bör forskaren sträva efter objektivitet, vilket uppnås i högre utsträckning genom att hålla en distans till respondenten. I kvalitativa, hermeneutiska metodansatser understryks vikten av inlevelse och subjektivitet, vilket skulle innebära att närhet till respondenterna t.o.m. kan vara önskvärd.

Distansperspektiv kan också förekomma i kvalitativa studier, en syn som kan representeras av följande citat:

"...we would recommend that researchers choose settings in which the subjects are strangers to them and in which they have no particular professional knowledge or expertise." (Bogdan & Taylor 1972:28)

Distans i denna bemärkelse innebär alltså att forskaren dels inte skall ha några andra relationer än de rent forskningsmässiga och dels inte skall ha någon kännedom om respondenten utöver vad som kan vara relevant för datainsamlingen. Utöver detta skall forskaren inte ha någon yrkesmässig kunskap om respondentens omgivning. Det rekommenderade förhållningssättet skall vara en garanti för att tolkningar inte påverkas av irrelevanta hänsyn.

Närhetsperspektivet kan representeras av följande två citat:

"Med access avses möjligheterna för forskaren/konsulten att få tillträde till data och information om det han studerar." (Gummesson 1985:19)

Repstad skriver i samma anda:

"Även om inte alla forskningsprojekt på långt när är sådana att forskaren lever flera år bland sina data, är det ändå ett centralt ideal i kvalitativ metod att så långt som möjligt ha ett nära och direkt förhållande till det man studerar." (Repstad 1988:9)

Närhetsperspektivet innebär att forskaren bedöms få tillgång till mer relevant information än om han distanserar sig från respondenten och hans omgivning. I detta ligger ett antagande om att relationer av olika slag ger tillgång till information som annars inte skulle bli tillgänglig. Gummesson redovisar ett exempel där en respondent inte velat berätta om hur ett beslut egentligen fattades därför att tillvägagångssättet kunde betraktas som oseriöst. Med personliga relationer till respondenten kan enligt närhetsperspektivet bl.a. denna typ av felkällor undvikas.

Likaväl som det kan finnas fördelar med att samla data via en roll med ett närhetsperspektiv kan det finnas nackdelar. Gummesson nämner *selektiv perception* som en risk i sammanhanget. Gummessons exempel utgår från en situation där forskaren även agerar konsult. Med selektiv perception menas att forskaren/konsulten

"...ser ett företags problem genom olika glasögon som begränsar synfältet till vissa företeelser." (Gummesson 1985:39)

Hur garderar man sig mot selektiv perception? Även om man är medveten om risken för detta är det ju inte säkert eller ens troligt att man helt kan undanröja risken för detta. Som forskare kan man dra fördel av det vetenskapliga samhällets granskande funktioner. Genom seminarier har avhandlingens kapitel granskats, vilket kan ses som en väg att erhålla en mångsidig perception, vilket i sin tur bidrar till ett värdefullt resultat.

En annan risk som Gummesson nämner, är *blockerande förförståelse*. Med detta avses

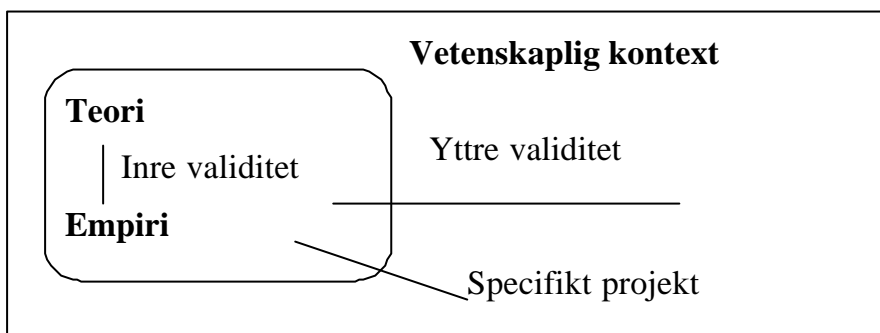
"...kunskaper och förhållningssätt som är starkt selektiva p.g.a. att man är bunden till ett bestämt paradigm och de teorier modeller, checklistor etc. som emanerar ur detta. Kunskapen kan vara detaljrik och omfattande men fungera som en fördom: man fortsätter att gräva i samma hål, man blir 'fackidiot' och försöker pressa in verkligheten i begrepp och teorier som gör våld på verkligheten istället för att hjälpa oss att förstå den." (Gummesson 1985:39)

Att helt gardera sig mot blockerande förförståelse tycks som en omöjlighet. Varje tänkt begrepp innehåller ju på sätt och vis ett ställningstagande och i förlängningen en fördom. Att fortsätta att studera t.ex. kalkylsystem behöver dock inte automatiskt innebära att det är ett uttryck för vare sig blockerande förförståelse eller selektiv perception, i synnerhet inte om man låter resultatet av studierna underkastas den akademiska samhällets kritiska mekanismer. En omständighet som förstärkt denna mekanism är det faktum att granskningen till största delen utförts av personer, som i icke ringa utsträckning tillhört ett paradigm med delvis annat synsätt på systemutveckling, t.ex. vad gäller metodanvändning.

I de aktuella fallen har närhetsperspektivet varit det rådande. Jag har vid intervju tillfället haft kännedom om såväl respondenterna som de miljöer de verkat inom. I själva verket har min kännedom om personer och organisationer varit utgångspunkten för valen. Motiven redovisas i samband med redovisning av respektive projekt i *Del II*. Respondenterna och deras organisationer är ej anonymiserade.

2.6.5 Datainsamling och validitet

Validitet är en egenskap hos datainsamlingsmetoder som berör riktighet i presenterade forskningsresultat. Hög validitet hos en datainsamlingsmetod innebär frånvaro av systematiska (mät)fel. Svenning (1996:60) anför validitetskriterier som väsentliga vid val av undersökningsdesign (se Figur 20 nedan). *Inre validitet* handlar om huruvida urval av respondenter, formulering av frågor etc. kan anses vara rimligt med avseende på projektets syfte. Inre validitet handlar följaktligen om forskningsmetodens interna konsistens. Mäts rätt sak? *Yttre validitet* handlar om forskningsprojektet som sådant och huruvida det medges möjligheter att dra relevanta slutsatser utifrån de i projektet ingående studierna. Yttre validitet handlar följaktligen om generaliseringsförmåga. Är resultatet överensstämmande med verkligheten.



Figur 20 Förhållandet mellan inre och yttre validitet (Svenning 1996:62).

När det gäller denna avhandling är det specifika projektet kalkylprogramanveckling. Den inre validiteten berör relevansen i valet av teoretiskt underlag och de empiriska undersökningar som gjorts. Exempel på validitetsfrågor för teoretiskt underlag är relevansen i valet av den praktikgeneriska modellen som struktur för referensramen och huruvida denna bidrar till att utveckla en anvecklargenerisk modell. Exempel på validitetsfrågor för empiriska undersökningar är relevansen i valet av djupintervjuer av anvecklare som observationsmetod. Dessa frågor diskuteras i detta kapitel samt i respektive empirikapitel. Den vetenskapliga kontexten är det sammanhang i vilket projektet kalkylprogramanveckling har relevans. Projektet kalkylprogramanveckling tillhör en samhällsvetenskaplig tradition. Det som studeras är människor och deras handlande, instrument för handlande och resultatet (produkt) av handlandet, samt vilka förutsättningar och effekter som kan förknippas med dessa aktiviteter. Projektet drivs organisatoriskt inom ramen för forskningsämnet *Informationssystemutveckling*, men får anses som i viss utsträckning ämnesöverskridande i och med att resultaten förväntas vara relevanta t.ex. för intressenter, som berörs av frågor kring styrande och genomförande av arbetsuppgifter i organisationer oberoende av IT-stöd. För Informationssystemutveckling finns relationen, i det faktum att anveckling är en form av systemutveckling, samt en form av användning av utvecklade informationssystem. Den yttre validiteten handlar om huruvida forskningsprojektet kan bidra till kunskapsutveckling inom ramen för den vetenskapliga kontexten. Syftet med avhandlingen är att bl.a. att formulera en begreppsmodell för KPA, en anvecklargenerisk modell, vilket kan ses som en form av generaliserande syfte. Diskussion om det relevanta i projektet förs i kapitel 1.1.

DEL II

DEL II

Empiri

"När man hittar redskapen för att analysera, då märker man hela tiden vad det är som är fel. På så sätt så förändras ju förutsättningarna. Och rätt som det är har man kommit till ett stadium, att nu kan vi inte bara köra det här med den organisation som vi har. Eftersom vi ser att det här inte är som vi trodde. Så då måste vi göra om organisationen. Det är ju så det blir hela tiden. Det glider hela tiden. Det nya är förändring. Det är det som är grejen. Man skall uppnå det optimala. Det hinner man aldrig till. Det är som att skjuta på rörligt mål hela tiden."

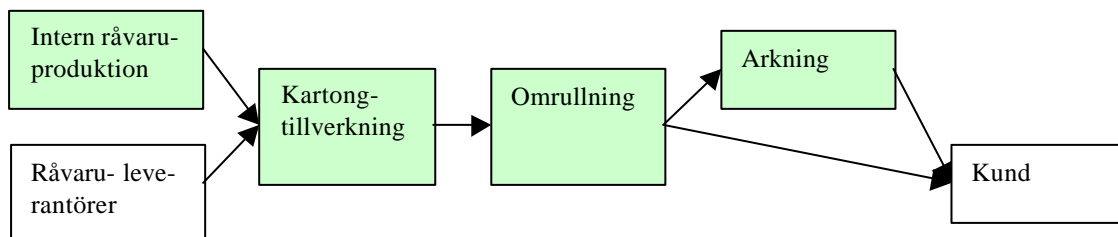
Bosse Magnusson, controller, AssiDomän Frövi (Avdic 1999:79)

- *Detta kapitel syftar till att redovisa tre personers förutsättningar för och effekter av anvecklingsverksamhet, med speciell fokus på möjligheten att ta hänsyn till verksamhetskunskap och svårformaliserbara mål.*

3 Industri

3.1 Företaget

Företaget där studien gjorts, AssiDomän Frövi, är en processindustri med över 700 anställda och en omsättning på ca två miljarder kronor per år. Företaget tillverkar kartong av två kvaliteter, vilka i sin tur tillverkas i olika ytvikter. Produktionen pågår i treskift året runt. En del av råvaran till kartongproduktionen tillverkas av företaget i en egen pappersmassafabrik. Övrig råvara köps från olika leverantörer. Tillverkade kartongtambourer¹ rullas om i en rullmaskin till mindre rullar. En del av produktionen arkas till mått som kunder beställt. Arkningen sker antingen i företaget i någon av de två arkmaskiner som finns där eller så sker arkningen externt i Sverige eller i Europa. Företagets produktion exporteras över hela världen med Europa som främsta exportområde. Företagets produktion visas schematiskt i Figur 21 nedan.

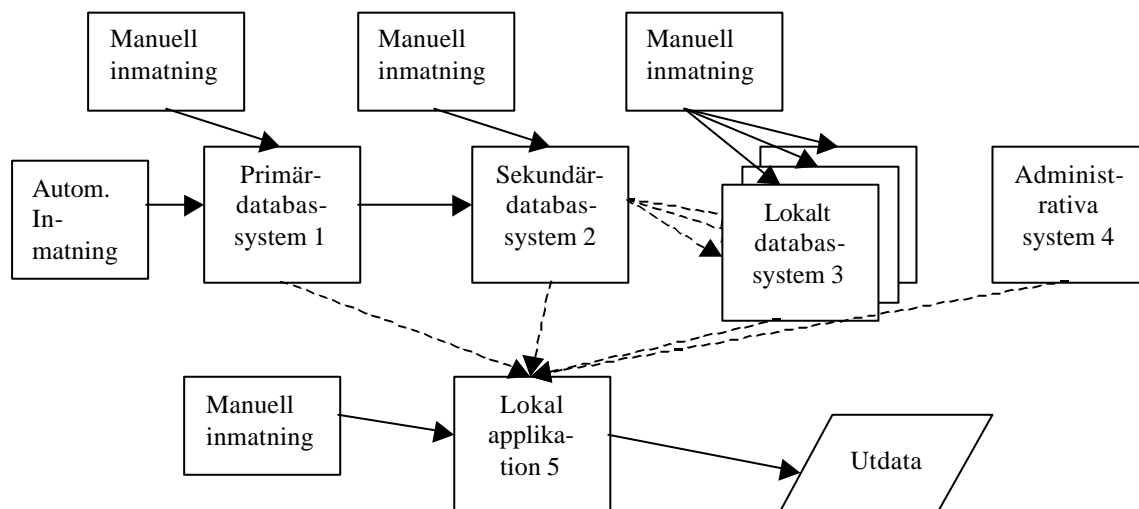


Figur 21 Schematisk beskrivning av produktionen på det studerade företaget.

¹ En tambour är en ca 6,5 meter lång och ca tre meter hög kartongrulle.

På företaget finns flera integrerade datasystem. I själva produktionen finns ett avsevärt antal processororienterade datasystem av olika slag, vilka inte behandlas vidare här. De system som är intressanta ur ett utvecklingsperspektiv, är administrativa system som ger möjlighet till analys av produktionsdata, transportdata, administrativa data² e.d.

De administrativa systemen med anknytning till produktionen bygger på data som lagras i ett produktionsdatasystem, här kallat primärdatan. Primärdatan innehåller produktionsdata som registreras automatiskt eller manuellt i direkt anslutning till produktionens olika led (se Figur 21 ovan). Primärdatan är upplagd som en relationsdatabas som kan nås med SQL-frågor från olika applikationer. Från primärdatan görs ett urval av information som ligger som rapporter i ytterligare en databas, sekundärdatan. Sekundärdatan innehåller förutom produktionsrapporter från primärdatan ytterligare produktionsinformation som matas in av användare. Även sekundärdatan är tillgänglig för SQL-frågor från olika applikationer. Informationen i sekundärdatan är dessutom tillgänglig i ett internt rapportsystem (intranät) som kan nås av de anställda via det lokala nätverket. Ytterligare information finns lagrad i ett antal databaser tillhörande lokala system av olika storlek och karaktär. Utöver dessa system finns standardsystem för ekonomiadministration, orderhantering, materialhantering, lagerhantering, laboriemätningar etc. Schematisk beskrivning av företagets datasystem visas i Figur 22 nedan. På företaget finns även ett lokalt nät med ca 200 PC-datorer sammanknutna. Alla användare har tillgång till programpaketet Office 97 där kalkylprogrammet MS Excel 97 ingår.



Figur 22 Schematisk beskrivning av de administrativa systemen på det studerade företaget. (Streckad linje = ev överföring)

3.2 Genomförande

För allmänna principer för metodanvändning hänvisas till kapitel 2, i detta avsnitt redovisas endast delprojektspecifika metoder.

² Med administrativa data avses här data som produceras av administrativa avdelningar, som t.ex. ekonomiavdelningen, inköpsavdelningen eller personalavdelningen.

3.2.1 Metod

Undersökningen som genomförts, kan betecknas som en kvalitativ fallstudie genomförd med (1) öppna intervjuer, (2) studier av kalkylsystem samt (3) informella samtal, som genomförts vid varierande tillfällen.

Intervjuerna (1) har följt intervjumallen i *Bilaga 2*. Intervjumallen innehåller allmänna frågor som syftar till att fånga yttre omständigheter kring anvecklarens arbetssituation. Större delen av intervjuerna har ägnats åt demonstration av och diskussion kring olika kalkylsystem. Intervjuerna är inspelade på band och även nedskrivna manuellt vid intervjutillfällena. Registreringen har gått till så att anteckningarna först renskrivits med hjälp av ordbehandlare därefter har intervjuanteckningarna kompletterats utifrån bandinspelningarna. När registreringen varit klar har upprepade genomläsningar genomförts för att koda innehållet. Kodningen har gått till så att noteringar gjorts i dokumentet, men skilt från den löpande texten med intervjuregistreringen. Åtskillnad mellan texttyperna har gjorts med hjälp av ramar eller indragningar. Antalet intervjuer per respondent har varierat mellan två (Thomas) och fem (Ismo). Respondenterna har fått reflektera på manus, ibland i omgångar, och fått kommentera och förändra. Det sista manusförslaget är intakt efter senaste revidering.

Studier av kalkylsystem (2) har gjorts på kalkylsystem som kopierats vid intervjutillfället eller vid något annat tillfälle t.ex. vid ett samtal. Studierna har gått till så att systemen har bedömts och provats med avseende på deras relevans för hur respondenterna kunnat uppfylla svårformaliserade mål i sin verksamhet med hjälp av kalkylsystemet.

Registreringar av de informella samtalen (3) har gjorts på det sätt som bedömts lämpligt vid eller efter samtalstillfället. Samtalen har förts direkt eller via telefon. Oftast har noteringar gjorts i anslutning till samtalstillfället på lappar som samlats för registrering, genomläsning och sammanställning på samma sätt som intervjuerna. Någon gång har noteringen gjorts vid ett helt annat tillfälle då något fenomen i omgivningen utlöste en association som resulterat i en reflexion med anknytning till respondentens verksamhet och kalkylprogramanveckling. Någon loggbok över hur många och hur långa samtal som förts har inte gjorts. En uppskattning är att antalet samtal, räknat över en femårsperiod kan vara ca 20 till ca 50 per respondent.

Ej refererade citat i texten är avskrifter från bandinspelningarna.

3.2.2 Urval av respondenter

Samtliga respondenter har valts därför att jag, utifrån min kännedom om deras verksamhet och deras uppfattningar, bedömt att de varit representanter för en grupp personer som använt kalkylprogram (och i ett fall även ett annat anvecklarverktyg) för att uppfylla verksamhetsetiska mål. Var och en har haft olika uppgifter i olika delar av

organisationen. Deras anvecklingsverksamhet har tagit sig olika uttryck, vilket beskrivs nedan.

3.2.3 Disposition av presentation

Presentationerna av respondenterna och deras erfarenheter följer följande disposition:

- beskrivning av respondenternas arbetsuppgifter och verksamhetskunskap
- redogörelse för verksamhetsmål ur respondentens perspektiv och professionella mål
- beskrivande presentation av respondentens kalkylprogramanvändning
- beskrivning och analys av respondentens kalkylprogramanveckling i termer av förutsättningar och effekter
- beskrivning av respondentens arbetssätt och relaterade aspekter
- sammanfattning

3.3 Ismo - produktionsplanerare

3.3.1 Uppgifter och verksamhetskunskap

Ismo har arbetat ca tjugo år på företaget och sedan åtta år som produktionsplanerare på kartongfabriken. Han är dessutom ansvarig för sk externarkning (se Figur 21 ovan). Under en period har han också arbetat till 75% som delprojektansvarig i ett OML-projekt³ under avdelningen planering/logistik. Förutom kalkylprogram använder Ismo ett standardprogram för trimningsdelen (se nedan) av produktionsplaneringsuppgifterna.

3.3.2 Mål

I Ismos fall är det svårt att skilja på verksamhetsmål och professionella mål. I och med att Ismo sysslar direkt med produktionen, utan att direkt tillämpa någon specifik profession, sammanfaller de båda målen. Ismos professionskunskap är produktionen på företaget (och på företag med liknande verksamhet).

Det av Ismo uppfattade målet med **produktionsplaneringen** är att optimera produktionen i kartongmaskinen. Optimeringen grundar sig på

- **Planering av användning av kartongmaskinen**, vilket innebär att planera in kundorder så att (1) kunderna får sina leveranser i tid samtidigt som (2) så lite maskintid som möjligt går åt till omställning. I denna syssla ingår även trimning (finplanering) för att utnyttja hela maskinens bredd.⁴
- **Planering av råvarutillgång**, vilket innebär samordning med leverantörer. Planeringen innebär att balansera tillgång mot lagerkostnad. Råvaror skall alltid finnas tillgängliga för produktionen samtidigt som för mycket kapital inte får bindas i rå-

³ Order Marknad Lager.

⁴ Maskinens bana är 6,75 meter bred, 10-12 cm på varje sida går inte att använda, resten skall utnyttjas.

varulager. Samordning med leverantörer sträcker sig normalt ett halvår fram i tiden.

Det av Ismo uppfattade målet med **externarkningsplaneringen** är att optimera arkningen. Optimeringen grundar sig på

- **Kapacitetsberäkning** av arkning på olika platser
- **Fördelning** av arkning utifrån **volym**
- **Fördelning** av arkning utifrån **destination**

Ismo ser två tänkbara huvudstrategier för planeringsarbetet, *produktionsorientering* eller *marknadsorientering*. Enligt en produktionsorienterad strategi körs långa serier i kartongmaskinen, vilket å ena sidan kan ge en jämn och säker produktion med god kvalitet men å andra sidan lagerproblem och låg kundanpassning. Enligt en marknadsorienterad strategi produceras kartong i cykler för att anpassa till kundorder, vilket kan innebära problem för produktion, p.g.a. omställningsarbete. Huvudstrategierna kan betraktas som alternativa *verksamhetsmål* ur produktionsplanerarens perspektiv.

Faktorer som påverkar planeringsbeslut är

- Kundefterfrågan
- Maskinutnyttjande kartongmaskin
 - Kvalitet
 - Volym
 - Bredd ("kan tummas på om kvaliteten kräver det, den är minst viktig")
- Utnyttjande av arkmaskiner
- Lager
- Transportkostnader

Kvalitetsparametern bestäms av kundefterfrågan. Enligt Ismo är det kundefterfrågan som i första hand styr planeringen.

Planeringen sker efter en produktionskedja, ytvikter som följer efter varandra måste köras i tur och ordning. Maskiner av den här storleken⁵ går inte att köra "...som om man producerade fiskerullar. Man kan inte köra på ena stunden och bromsa den andra. Det måste följa en viss lunk."

3.3.3 Anveckling, frekvens och typ

Ismo använder kalkylprogram varje dag och ungefär halva sin arbetstid. Utbildningsbakgrund är fyra heldagskurser för ca 4-5 år sedan. Ismo anser att kurser visar möjligheter medan praktisk tillämpning ger mer kunskaper. De kalkylsystem som Ismo bygger är av typen *Lilla kalkylsystemet* (se kap 8.3 för klassificering av kalkylsystem). Ismo har även initierat och deltagit aktivt i konstruktionen av ett flertal kalkylsystem av typen *Applikationen*, vilka bl.a. används för rapportering av råvaruåtgång och -lager, kontroll av leveranser, redovisning av planering av körcykler. Dessa kalkylsystem diskuteras inte vidare här.

⁵ Maskinen är 300 meter lång, tio meter bred och femton meter hög.

3.3.4 Anveckling, exempel och analys

Som skäl till varför kalkylprogram är användbart menar Ismo att "man får en exakthet, en noggrannhet i det man söker". Vidare: "Man kan söka vad som helst, kostnader, mätnivåer, tidsaspekter, man kan göra sorteringar." "Det poppar upp områden där man kan använda kalkylprogram." På den avdelning där Ismo arbetar händer det att han hjälper andra att utforma kalkylsystem för summeringar eller andra beräknande sammanställningar. Ismos arbete som produktionsplanerare har förändrats i och med tillgången till kalkylprogram. Kalkylprogram kan ersätta manuell hantering av:

- utseende, layout
- tydliga siffror
- summeringar
- procent
- avrundning
- "Allt. Ja inte riktigt allt."

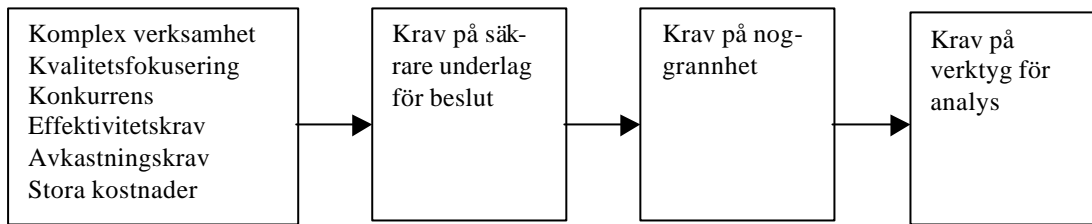
Ismo har arbetat på två avdelningar, produktionsplanering och planering/logistik. På produktionsplaneringsavdelningen mättades kalkylsystembehovet så småningom. Under tiden rationaliserades sekreteraruppgifter bort. Sekreteraranvändandet var ett slags dubbelarbete. På den nya avdelningen är PC-mognaden inte stor. Ismo ser även här behov av analys och uppföljning. Bl.a. finns ett behov av att följa upp s.k. externmarkning. Man måste veta vad det kostar per ton att arka och vad varje leverans kostar.

Noggrannhet

Ismo karaktäriserar kalkylprogrammet som ett flexibelt verktyg i planeringsarbetet. Det som han särskilt poängter är *noggrannhet*. Enligt Ismo utmärks den bransch där han arbetar av erfarenhet och tumregler. Förekommande uttryck som speglar detta är "Det borde vara...", "Så här är det...", "Det kommer att ta ungefär så här lång tid...". I själva verket är det svårt att göra exakta förhandsuppskattningar av den komplexa verklighet som företagets tillverkning innebär. Ismo menar att man med kalkylprogram kan få en bättre noggrannhet "om man lägger ner möda och följer upp". Flera skäl finns till att det ställs större krav på optimal planering. Några är: komplex verksamhet, ökad fokusering på kvalitet, konkurrens på en föränderlig marknad, krav på effektivitet och avkastning. Av dessa skäl är det viktigt för företaget att minska kostnaderna.

"På råvarusidan är det viktigt att få in rätta värden. Det handlar om stora pengar och därmed kostnader."

- Kalkylprogram kan användas som ett verktyg för flexibel planeringsverksamhet
- Planeringsavdelningen har med kalkylprogram fått en mer styrande roll än tidigare
- Besluten är bättre nu
- Visst rationaliseringsbehov kan tillgodoses med kalkylprogram
- Om beslut skall fattas med säkert underlag ställs det krav på noggrannhet i beslutsunderlaget. Detta skiljer sig från en tradition av tumregler och erfarenhetsmässig uppskattning (se Figur 23 nedan).

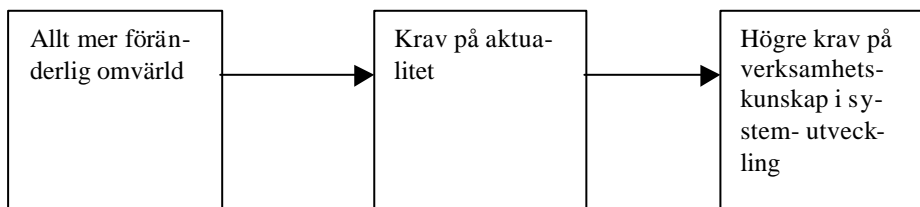


Figur 23 Krav på säkrare beslutsunderlag ger krav på noggrannhet ger krav på verktyg som stödjer detta.

Aktualitet, föränderlighet och rimlighetsbedömning

För att kunna planera med noggrannhet menar Ismo att man inte skall "gräva för mycket bakåt i siffror". Siffror skall ha hög *aktualitet*. Eftersom omvärlden är alltmer föränderlig tillgodoser inte befintliga historiska data de behov som finns. I en föränderlig värld ändras produktionsmixen, nya kvalitetskrav formuleras, kundsammansättning förändras, uppfattningar om mätmetoder förändras etc. Dessa förändringar innebär tillsammans att analyser som skall vara aktuella inte säkert kan utnyttja befintlig information, som ligger lagrad i databaser, som planerades under andra villkor än vad som gäller vid planeringstillfället. Ismo: "Man skall jobba med det senaste. Det är mer föränderligt än det har varit. Det gör att nya värden behövs."

Ett exempel på hur föränderlighet kan påverka verksamheten är när en kund har en stor del av den totala produktionen. När kundens produktmix förändras, ligger olika kvaliteter inte alls på samma nivå som tidigare, vilket innebär nya förutsättningar för planering (se Figur 24 nedan). Ett annat exempel på att man kan inte titta för mycket bakåt vid analys, är om företaget tappar en större kund, och måste ersätta denna med flera småkunder. Befintlig produktionsstatistik bygger då på en annan situation än den som man skall planera för. Ett tredje exempel är när kundefterfrågan och produktmixen styr verksamheten. I en given maskin kan lägre produktion vara lönsam om vissa produkter produceras istället för andra. Man producerar vid en maskin mindre men mer lönsamt idag än tidigare. Detta beror på olika kunder är olika lönsamma för produktionen totalt.



Figur 24 En allt mer föränderlig omvärld ställer krav på verksamhetskunskap i systemutvecklingsarbetet.

Ismos strategi för insamling av planeringsunderlag i en föränderlig omvärld följer följande tre steg:

1. Vad behöver vi veta? Vilka data?
2. När mallen är klar, mät!

3. Efter tre månaders mätande (minst) kan mallen användas. "Då kan man börja skapa statistik"

En annan typ av aktualitetsbehov än den som berör behovet av aktuell information i en föränderlig verksamhet är aktualitet i problemlösningssituationen på så vis att problem och/eller möjligheter skall kunna hanteras då de är aktuella. Tillkomsten av ett kalkylsystem för **allokering av restpallar** kan sägas ha fungerat på det sättet.

Företaget köper engångspallar till order. Dessa kostar ca en viss summa per m². Av olika skäl (orderförändring, orderannullering etc.) blir en del av pallarna kvar på företaget. Kalkylsystemet gör det möjligt att kolla om det finns sk restpallar kvar i lager som ett alternativ till att köpa nya.

Systemet (se Figur 25 nedan) består av en lång lista med befintliga restpallar. Ovanför listan finns ett beräkningsområde, som föreslår lämplig pallstorlek utifrån visst arkformat. Användaren matar in arkbredd och arklängd, varefter systemet beräknar hur stora pallarna skall vara. I listan kan man sedan se om det finns några pallar i rätt storlek. Listan måste sorteras före varje användning. Kalkylsystemet kräver kontinuerlig registrering av restpallar.

	A	B	C	D	E	F	G	H	I	J	K
1		Beställt arkformat		förslag på pallstorlek: enkelstapling				förslag på pallstorlek: dubbelstapling			
2		700 x	1000 mm	min bredd	70,5 cm	min längd	100,5 cm	min bredd	100,5 cm	min längd	140,5 cm
3				max bredd	75,0 cm	max längd	105,0 cm	max bredd	105,0 cm	max längd	145,0 cm
4		Pallformat:						kan användas till arkformat:			
5	nr	bredd	längd	antal	palltyp	order nr	lagerplats	min bredd	max bredd	min längd	max längd
6	1	40,0 cm	60,0 cm	10		1 22222-1		35,0 cm	39,5 cm	55,0 cm	59,5 cm
7	2	47,0 cm	68,0 cm	8		2 31818-1		42,0 cm	46,5 cm	63,0 cm	67,5 cm
8	3	48,5 cm	83,0 cm	6		2 30984-1		43,5 cm	48,0 cm	78,0 cm	82,5 cm
9	4	49,0 cm	90,5 cm	72		2 28877-1		44,0 cm	48,5 cm	85,5 cm	90,0 cm
10	5	50,5 cm	70,5 cm	10		2 29667-1		45,5 cm	50,0 cm	65,5 cm	70,0 cm
11	6	51,5 cm	74,5 cm	50		1 22402-1		46,5 cm	51,0 cm	69,5 cm	74,0 cm
12	7	52,0 cm	76,0 cm	4		2 18682-1		47,0 cm	51,5 cm	71,0 cm	75,5 cm
13	8	52,5 cm	83,0 cm	16		2 29852-1		47,5 cm	52,0 cm	78,0 cm	82,5 cm

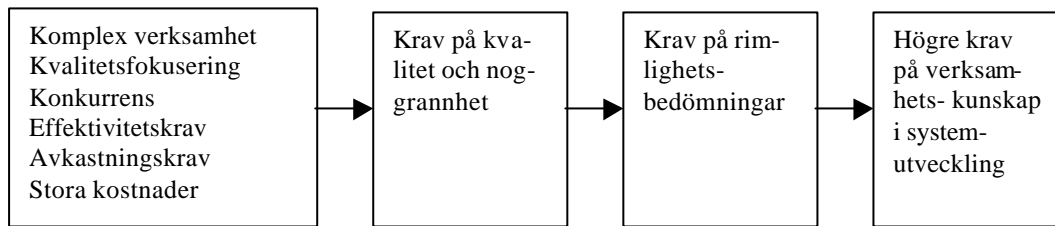
Figur 25 Kalkylsystem för allokering av överblivna pallar.

Första veckan som systemet användes hade man 1400 pallar i lager. På en vecka fick man ner det till 900, vilket spara pengar både i form av att kostnader undviks och att lagerplats friläggs. Systemet är udda men viktigt. Det skulle vara svårt att införliva i standardssystem. Det kom till som en idé som förverkligades för att lösa ett plötsligt uppmärksammat problem.

- System kan uppstå ur ett infall, som uppstår i det dagliga arbetet
- Viktigt att kunna förverkliga idén på en gång
- Kalkylprogram kan tillgodose aktualitet i hantering av idéer och problemlösning

Ismo menar att han inte skulle kunna nå samma effektivitet utan kalkylprogram. Han tycker att han nu snabbt kan få fram korrekta värden och uppgifter för att uppfylla krav på hög informationskvalitet för att stödja en komplex och kapitalintensiv verksamhet. "Det blir mer rätt nu. Den som bygger måste ha ett rimlighetstänkande. Är värdet jag fått fram rätt? Det är viktigt att kunna laborera och bedöma data på olika sätt, t.ex.

värden per timme." Rimlighetsbedömningar kräver verksamhetskunskap (se Figur 26 nedan).



Figur 26 En komplex och kapitalintensiv verksamhet leder till behov av verksamhetskunskap vid kalkylprogramutveckling.

Användning av kalkylprogram ger säkerhet i beslutsfattande. Ismo tycker sig inte behöva tveka eller tro att det blir galet p.g.a. felbedömning.

"Jag kan göra avrop på förbrukningsvaror. Vi kan säga vi behöver handla för 20 Mkr. Det här behöver vi. Jag kan då stoppa planerade leveranser för lika mycket tacka vare egna kalkylsystem. Dessa egna system kommer inte att ersättas av planerade standardsystem. Standardsystemet kommer bara att jobba i produktionskedjan."

Ismos kalkylsystem skulle kunna implementeras i ett egenutvecklat centralt system eller i ett standardsystem. Skälen till att detta inte görs är två. Dels är det en kostnad som bedöms vara större än om Ismo bygger systemen själv. Dels finns det en risk att det nästa vecka skulle "poppa upp" nya idéer och behov.

Uppföljning, kontroll och verifiering

Ett av Ismos kalkylsystem fungerar som en **produktionsuppföljning**. Systemet innehåller kvantiteter för företagets externarkning fördelat på ytvikter och arkare. Beräkningen i systemet är summering av kvantiteter som registreras på blankettliknande blad. Syftet med systemet är budgetuppföljning/vecka. Företaget måste köpa externa arkningstjänster, bl.a. för att följa med marknadens utveckling. Om t.ex. Englandsmarknaden ökar, ökar arkningen i England.

Bakgrunden till systemet är en diskussion kring värdet av de data som använts för planering. Någon hävdade att planeringen byggde på felaktiga data. Ismo ville undersöka hur verkligheten förhöll sig.

"Jag ville skapa fakta om detta"

"Det skall inte finnas tveksamhet"

"Folk skall inte påstå saker"

"Det skall vara sanning"

Kalkylsystemet utgörs av en arbetsbok med ett blad per externarkare (se Figur 27 nedan) samt ett summeringsblad (se Figur 28 nedan). Registreringsarbetet utförs av sekreterare på Ismos avdelning. Systemet ersätter ett manuellt blankettsystem. En effekt är att sekreterarna tar del av betydelsen av vad de sysslar med. I tidigare system såg sekreterarna bara ett konto för externarkning och en summa pengar. Man kunde aldrig se hur mycket en speciell arkare kostade. Nu kan man se och snabbt få fram siff-

Användare och utvecklare

ror fördelade på kvalitet, ytvikt och arkare. Systemet ger även noggrannhet mot varje arkare när man tar kontakt med dem. Ismo har utvecklat systemet och finns till hands vid eventuella problem. Systemet kan därför ses som ett uttryck för *fadderanveckling* (se kap 11.2).

	A	B	C	D	E	F	G	H
1	Förbrukningsregistrering - Externarkning							
2		210 g/m ²		230 g/m ²		250 g/m ²		27
3	Order nr	förbrukat, kg	utvägt ark, kg	förbrukat, kg	utvägt ark, kg	förbrukat, kg	utvägt ark, kg	förbruk
4								
5								
6								
7								

Figur 27 Del av blad för viss externarkare i kalkylsystem för uppföljning av externarkning.

	A	B	C	D	E	F	G	H	I	J	K
1	Förbrukningsrapport - Externarkning										Vecka
2	Förbrukade rullar						Netto utvägt(prima)ark				
3	g/m ²	Arkare 1	Arkare 2	Arkare 3	Övriga	Summa	Arkare 1	Arkare 2	Arkare 3	Övriga	Summa
4	210 g/m ²	0				0 kg	0				0 kg
5	230 g/m ²	0				0 kg	0				0 kg
6	250 g/m ²	0				0 kg	0				0 kg
7	270 g/m ²	0				0 kg	0				0 kg
8	290 g/m ²	0				0 kg	0				0 kg
9	310 g/m ²	0				0 kg	0				0 kg
10	330 g/m ²	0				0 kg	0				0 kg
11	350 g/m ²					0 kg					0 kg
12	370 g/m ²					0 kg					0 kg
13	390 g/m ²					0 kg					0 kg
14	410 g/m ²					0 kg					0 kg
15	430 g/m ²					0 kg					0 kg
16	450 g/m ²					0 kg					0 kg
17	Summa	0 kg	0 kg	0 kg	0 kg	0 kg	0 kg	0 kg	0 kg	0 kg	0 kg

Figur 28 Summeringsblad i kalkylsystem för uppföljning av externarkning.

Kalkylsystem kan fungera som *kontroll* och *verifiering* av befintliga rutiner. Då Ismo gjorde ett kalkylsystem för **beräkning av arkantal per pall** (se Figur 29 nedan), visade det att resultatet avvek från en annan beräkning. Personer med god verksamhetskunskap har i vissa fall större möjligheter att veta vad som är rimligt och rätt. Då förhållanden av något slag förändras, t.ex. förändrad produktmix, nya produkter eller förändrad produktion av befintliga produkter, skapas ett behov av kontinuerlig kontroll för att garantera systems funktionalitet.

Då information presenteras på ett nytt sätt, t.ex. genom högre detaljeringsgrad än normalt eller genom ny indelning av materialet kan detta bidra till att problematisera och belysa problemområden. Även om det inte är ny information, kan en presentation, grafisk eller textuell, leda till nya inblickar i en välbekant verksamhet. Ismo konstruerade ett kalkylsystem med anledning av diskussion om omläggning av körcykler på kartongfabriken. Systemet visade vilka effekter körcyklerna fick på den efterföljande arkningen. Via systemet kunde man analysera hur länge man behövde köra för att produktionen skulle räcka fram till nästa arkning. Resultatet blev att körcykeln skulle vara

Industri

	A	B	C
1	Antal ark per pall		
2	Order nummer	999999_9	
3	arkbredd,mm	800 mm	
4	arklängd,mm	1 200 mm	
5	ytykt	350 g/m ²	
6	beställd kvantitet,kg	1 500 kg	
7	beställt antal ark	4 464 st	
8	max pallhöjd	125 cm	
9	Antal pallar,enkel,- dubbelstapling	3 st	2 st
10	Antal ark/pall, enkelstaplat	1 488 st	
11	Antal ark/pall, dubbelstaplat	2 232 st	
12	nominell pallhöjd, enkelstaplat	93 cm	
13	nominell pallhöjd, dubbelstaplat	74 cm	
14	nominell pallvikt,enkelstaplat	500 kg	
15	nominell pallvikt, dubbelstaplat	750 kg	
16	Pallformat, enkelstaplat	80,5 x	120,5 cm
17	Pallformat, dubbelstaplat	120,5 x	160,5 cm

Figur 29 Kalkylsystem för beräkning av arkantal per pall.

20 dagar lång. Det blev 17 cykler per år. Ett resultat av presentationen av kalkylsystemets utdata blev en diskussion om vad som skulle styra verksamheten. Skulle arkningen gå hela tiden eller var viktigare att kartongmaskinen gick i längre cykler. Vilka effekter fick alternativen för t.ex. kvalitet, leveranssäkerhet och kostnader. Ismo betonar att det är viktigt att sätta sig in i olika parterers perspektiv och inte utnyttja producerad information på ett mekaniskt sätt.

Fördelar med kalkylprogram enligt Ismo:

- Kalkylsystemutveckling kan fungera som kontroll och verifiering av (dolda) beräkningsrutiner i system.
- Kalkylsystemet kan skapa en diskussion
- Olika val och deras konsekvenser blir genomskinliga
- Ställningstagandena hamnar på en högre nivå

	A	B	C	D	E	F	G	H	I	J	K	
1	Machine capacity									SC 1	SC 2	
2										machine speed	100 m/min	200 m/min
3										speed efficiency	80%	75%
4										time for order change	15 min	10 min
5	width x length	g/m ²	Vikt/1000 ark	quantity	number of sheets	number of meter	no rolls	time for roll change				
6	1 101 mm	600 mm	450 g/m ²	297,27 kg	1000 kg	3364 sh	2 018 meter	1 rl	sheeting time/singel rolls	40 min	23 min	
7	* AM 1 double rolls with sheet width < 1101 mm				* AM 2 double rolls with sheet width < 801 mm				sheeting time/double rolls	28 min	17 min	
8												

Figur 30 Kalkylsystem för simulering av arkningsfördelning.

Kalkylsystem kan ha ett *pedagogiskt syfte* (se även om presentation nedan). Ett kalkylsystem som Ismo gjort (se Figur 30 ovan) simulerar **fördelning av arkning** mellan de två arkmaskinerna som finns på företaget. Eftersom planering bygger på budget är det viktigt att budgetutformningen bygger på realistiska planer. "Om man vill diskutera budget underlaget kan man inte komma och säga: 'Fan vad det tar lång tid att arka det här'. Det kan tas personligt. Om man visar ett sånt här program så förstår man att långa ark tar lång tid kort ark går fortare. Här kan man få svar."

3.3.5 Arbetssätt

Nackdelar

En *nackdel* som Ismo påpekar är att det tar tid att bygga kalkylsystem. I början gjorde han "onödiga grejor". Presentationer kunde bli överarbetade. Ett annat problem är att få tid att förverkliga alla idéer. Det kan vara svårt att hitta rätt sätt att utforma kalkylsystemet så att det blir som man tänkt sig. Ytterligare ett problem som Ismo har haft är rörig organisation av kalkylsystemen/dokument på sekundärminne. Så småningom har han lagt upp ett katalogsystem som är indelat efter arbetsuppgifter.

Nackdelar med kalkylsystemutveckling enligt Ismo:

- Risk för överarbetning
- Svårt att få tiden att räcka till
- Katalogsystemet kan bli rörigt när det blir många dokument

Ändringsbarhet

Problem som Ismo upplevt vid kalkylprogramutveckling är om uppgifter i ett kalkylsystem skall kopplas till parametertabeller, för att minska ingreppen i själva kalkylsystemets formel. Ett annat är att göra kalkylsystemen generella på så vis att de anpassar sig efter indata. Ytterligare ett problem är felmeddelanden som kan se illa ut på presentationer.

- Koppla till kalkylblad till t.ex. parametertabeller⁶
- Villkorsformulering⁷
- Undertrycka felvärden eller andra värden⁸

Det viktiga med kalkylsystem är att de blir lätta och enkla och att det blir tydligt och lätt att förstå själv. Ismo fastnade förr vid formuleringen av rubriken. Skriver nu "vad som helst" för att komma förbi detta. Ismo försöker slippa skärmbildsrullning. "Det bör helst vara en bild per sida."

Ett användningsområde som Ismo använt kalkylprogram till är för *presentationsändamål*. Vissa kalkylsystem har Ismo gjort enbart för att presentera informationsunderlag, t.ex. inför ett beslut. Presentationen skall, enligt Ismo, poängtera viktiga saker. Dessa kan formateras med gråton eller fylligare text. Ett annat sätt att poängtera är att rama in summeringsraden, om sådan finns.

⁶ Kan i MS Excel ibland lösas med funktionen LETAUPP().

⁷ Kan i MS Excel ibland lösas med funktionen OM() t.ex. =OM(A1>0;"positivt";"0 eller negativt").

⁸ Kan i MS Excel ibland lösas med funktionerna OM(ÄRFEL()) t.ex. OM(ÄRFEL(F2);0).

Då Ismo med åren blivit allt händigare med kalkylprogrammet har han även gjort kalkylsystem till medarbetare med mindre kalkylprogramkunskap, vilket kan betecknas som en typ av *fadderanveckling* (se kap 11.2). En viktig aspekt är cellskydd då man gör kalkylsystem till medarbetare, som inte kan få tillbaka det de tar bort av misstag. "Systemet skall vara anpassat till kompetensen".

Hands-on

Ismos *arbetsätt* kan karakteriseras som "hands-on". "Jag ser behov och tar mig an dem. Jag börjar där jag tycker och fortsätter när det passar." Ett exempel på detta arbetsätt är ett system för **taraberäkning**⁹. Ismo beskriver:

"En av våra killar kom upp och sa: Polen kräver att vi deklarerar tara vid gränsen i kilo eller i pengar. Annars åker vi på en straffavgift på 1 % av fakturabeloppet som måste betalas vid passering av gräns. Men har vi dessa uppgifter med oss så slipper vi denna extrakostnad. Jag fick idén direkt. Det tog 15 till 20 minuter att göra."

Systemets funktion är simulering av emballagevikt (tara). Tara = omslagspapper, ytterdell, innerdell, hylsa och plugg. Antaganden i systemet är att omslaget är 40 cm bredare än rullen, och att det lindas två gånger runt rullen. Rondellen = ytan * ytvikt. Beräkning kan ske för rullar och pallar. I systemet utmärks inmatningsfält av skuggade celler (se Figur 31 nedan).

	A	B	C	D	E	F	G	H
1	Taraberäkning							
2	rullbredd	diameter	hylsdiameter	antal rullar	pallbredd	palllängd	pallhöjd	antalpallar
3	1 010 mm	1 750 mm	300 mm	15 st	850 mm	1 250 mm	1 150 mm	15 st
4	taravikt/rulle taravik/totalt			taravikt/pall taravik/totalt				
5	Totalt	16,9 kg	253,1 kg	Totalt			21,6 kg	323,7 kg
6	omslag	3,4 kg	51,1 kg	pall			20,2 kg	302,8 kg
7	yterdell	1,1 kg	16,2 kg	plast			1,4 kg	20,9 kg
8	innerdell	2,9 kg	43,3 kg					
9	hylsa	8,1 kg	121,3 kg					
10	plugg	1,4 kg	21,2 kg					

Figur 31 Kalkylsystem för taraberäkning.

Detaljeringsgrad

Ett val som systemutvecklare står inför är vilken *detaljeringsgrad* som skall finnas i ett system. Ett kalkylsystem som Ismo gjort syftar till **kostnadsjämförelse för arkning** (se Figur 32 nedan). Det visar kostnader för företagets arkning extern arkning månadsvis. Systemet har ett konkret syfte, att jämföra kostnader mellan olika arkare. Viktigt att den som följer upp är kunnig och kan välja lämpliga variabler och lämplig detaljeringsgrad utifrån syfte och målgrupp.

⁹ Tara är den del av en leverans som inte hör till själva produkten, t.ex. omslagspapper.

Användare och utvecklare

	A	B	C	D	E	F
1	ARKNING Totalt 1997				arkningskostnad	
2	ARKARE	Brutto ton	Netto ton	v.grad	totalt	/netto ton
3	Arkare 1	17 768,4 ton	16 116,8 ton	90,7%	7 416 962 kr	460 kr
4	Arkare 2	1 926,4 ton	1 724,8 ton	89,5%	807 860 kr	468 kr
5	Arkare 3	0,0 ton	0,0 ton	#Division/0!	0 kr	0 kr
6	Arkare 3	584,4 ton	558,0 ton	95,5%	226 144 kr	405 kr
7	Arkare 4	5 771,2 ton	5 443,2 ton	94,3%	2 312 808 kr	425 kr
8	Intern arkning	159 516,0 ton	149 312,0 ton	93,6%	66 214 052 kr	443 kr
9	Övriga	0,0 ton	0,0 ton	#Division/0!	0 kr	#Division/0!
10	Totalt	185 566,4 ton	173 154,8 ton	93,3%	76 977 825 kr	445 kr

Figur 32 Del av kalkylsystem för jämförelse av arkningskostnader (siffrorna är fiktiva).

Riktlinjer

Ismos riktlinjer för anveckling:

- Tydligt
- Enkelt
- En bild/skärm sida
- Formulera rubriken på själva systemet sist
- Överdriv inte utformning
- Poängtera väsentligheter
- Skydda celler om kalkylsystemet skall användas av andra
- Välj detaljeringsgrad och antal uppgifter efter syfte, sammanhang och målgrupp
- Var inte för detaljerad
- Ta inte med för många variabler

3.3.6 Sammanfattning - Ismo

I Ismos verksamhet är planeringsuppgifter lämpliga för anveckling eftersom det finns många mål, höga aktualitetskrav, krav på verksamhetskunskap och analysverktyg samt krav på relevanta och noggranna data.

Många mål

Kombination av olika mål gör det viktigt att vara föränderlig. Faktorer som påverkar produktionsplanering på företaget:

- Kundens krav och förväntningar¹⁰
- Kvalitet

¹⁰ Gäller speciellt då företaget verkar på en konkurrensutsatt marknad.

- Volym
- Komplex teknik
- Komplicerad produktionsprocess¹¹
- Kostnadskrävande produktion
- Transportkostnader och möjligheter¹²
- Råvarutillgång och -pris

Ovanstående faktorer är delvis motsatta. Volym och kvalitet är t.ex. faktorer som motverkar varann. Om det bara gällde att producera stora volymer utan hänsyn till kvalitet så vore planeringen enklare.

Aktualitet

En föränderlig värld och en komplex verksamhet gör behovet av snabb information viktigare, vilket leder till aktualitetskrav och krav på anpassning till mål, vilket i sin tur ökar kraven på föränderlighet i informationsförsörjning.

Verksamhetskunskap och analysverktyg

Avsaknad av något så när säkra bedömningsmallar skapar ett behov av fakta/empiriskt prövat underlag. Det är inte självklart vilka mått som skall användas för att mäta verksamhet, hur mätningar skall göras och om mätningarna är korrekta. Det finns ett behov av verksamhetskunskap och analysverktyg med vars hjälp analysverktyg kan skapas.

Relevanta noggranna data

Beslutsutveckling på säker grund, beräkningsverifiering, verksamhetsoptimering och verksamhetsutveckling kräver relevanta, väl underbyggda, noggranna data.

Noggranna underlag skapar diskussioner och överväganden på ett högre plan då fakta kan betraktas som korrekta. Ifrågasättande av underlag undviks, vilket gör att diskussionen kan koncentreras på framförda argument.

3.4 Bosse - controller

3.4.1 Uppgifter och verksamhetskunskap

Bosse har arbetat ca tjugo år på företaget som controller på ekonomiavdelningen. På företaget finns två controllerfunktioner¹³ en produktionsinriktad och en marknadsinriktad. Bosse innehar den förstnämnda.

I Bosses arbetsuppgifter ingår "traditionella basuppgifter, vilka är dels externa basuppgifter (rapporter uppåt till huvudkontoret och underlag till årsredovisning) och dels specialuppgifter (produktion av standardrapporter och uppdrag). Bosse är även ansva-

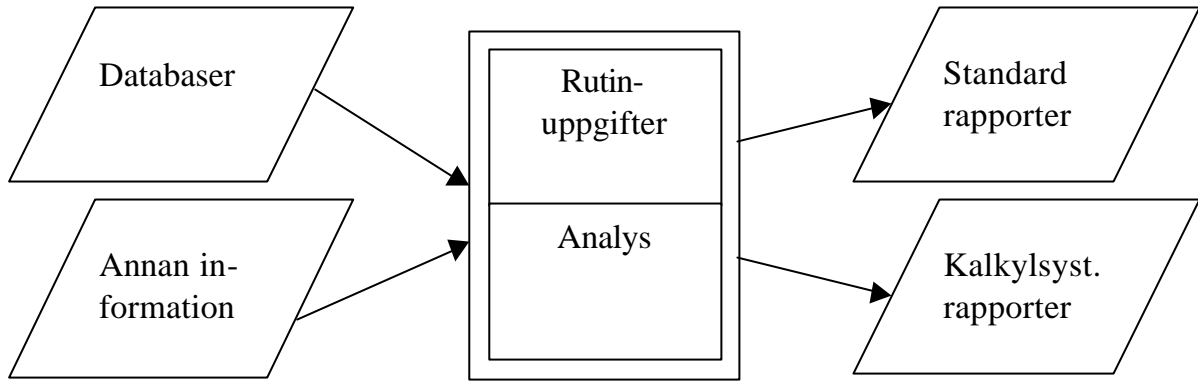
¹¹ T.ex. då produktionen sker i flera led, se Figur 21 sid 57.

¹² Gäller speciellt om produkter är av viss vikt och volym.

¹³ "controller = (eng.), befattningshavare inom företag och förvaltningar med uppgift att verka för att organisationens ekonomiska möjligheter tillvaratas på ett effektivt sätt vad beträffar resurser, kostnader och intäkter." (Nationalencyklopedin)

rig för budgetarbetet. Han driver sina uppgifter självständigt med de riktlinjer som finns på företaget som ram. Han har möjlighet att ta hjälp av medarbetare och ekonomichef.

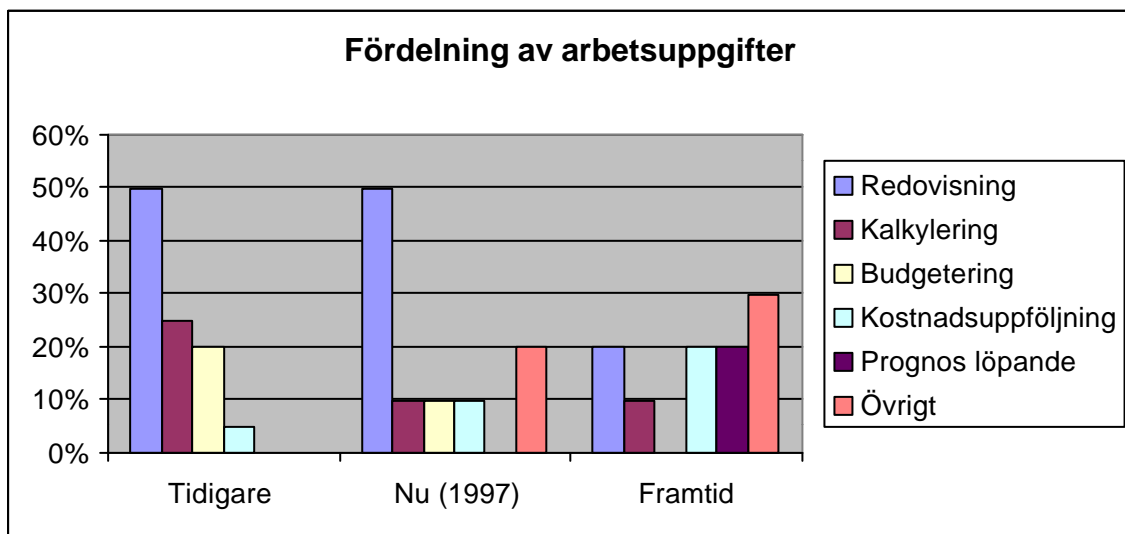
Såväl standardmässiga som speciella analyser och rapporter bygger på insamling av data från någon eller några delar av produktionen (se Figur 33 nedan). Informationskedjan på företaget beskrivs översiktligt i Figur 22 sid 58.



Figur 33 Schematisk beskrivning av Bosses informationshantering.

Rapporterna som produceras på ekonomiavdelningen lagras (i allt högre utsträckning) i en databas i det lokala nätverket där företagets personal kan nå den.

Arbetsuppgifterna på ekonomiavdelningen har förändrats och kommer att förändras (se Figur 34 nedan). Man räknar med att minska rutinarbetet, framför allt i form av redovisning och budgetering till förmån för mer flexibla arbetsuppgifter, för att kunna motsvara förändringar i organisationen och i omvärlden. Tidigare har man på ekonomiavdelningen inte haft någon tid över till analys, men genom utveckling av ett budgetsystem (se kap 6), förändring av rutiner samt genom implementering av ett nytt ekonomisystem har man frigjort och räknar man med att frigöra mer tid för analys (se Figur 34 nedan).



Figur 34 Fördelning av arbetsuppgifter på ekonomiavdelningen.

3.4.2 Mål

Företagets övergripande mål är *lönsamhet* och ekonomiavdelningens mål är att bidra till detta, "...att få folk att jobba med saker som vi tjänar på". Som ekonom gör man "...redskap, så att cheferna kan följa upp. Man har ansvar så att saker uppmärksammas." Ett medel för att uppnå lönsamhet är att förmedla lärande om ekonomiprocessen. "En ekonom ser inte så djupt men istället brett." Viktiga mål för företaget är också hög *produktionskvalitet* och *kundnytta*.

Verksamhetsmål:

- Lönsamhet
- Produktionskvalitet
- Kundnytta

Som ekonom är det viktigt *att ha integritet*. "Såna frågor är det hela tiden. Man kan bli påverkad från flera håll, men man måste välja det man tycker är rätt. Det kan t.ex. gälla priser och hur man skall räkna på en kalkyl."

Ett annat mål som Bosse har är att minimera tidsåtgången för utförande av standarduppgifter för att få *mer tid till analys*.

Många av de rapporter som produceras på ekonomiavdelningen konstrueras med andra rapporter som indata. Huruvida denna information är relevant och korrekt är en viktig uppgift för Bosse. *Att reflektera över informationens relevans, värde och korrekthet* kan ta sig många uttryck. T.ex. kan liknande information finnas på flera ställen, vilket gör det aktuellt att *jämföra informationen och värdera de mätmetoder* som använts för att producera denna. Dubbel förekomst kan ha för och nackdelar, såtillvida att den å ena sidan kan vara inkonsistent, vilket medför risk för felaktig information i analyser av verksamheten, det kan å andra sidan vara en fördel med dubbellagring då det ger möjligheter till kontroll och värdering av mätmetoder. *Databasernas innehåll* är betydelsefull för möjligheterna att utföra analyser. Kontrollrutiner kring *inmatning av data i databaserna* spelar roll för säkerheten i informationen. Det kan bli aktuellt att jämföra information för att värdera denna.

Professionella mål:

- Öka tidsandelen för analys relativt rutinuppgifter
- Att producera korrekta och relevanta rapporter
- Integritet
- Att förse beslutsfattare med redskap för att uppfylla verksamhetsmål
- Att operationalisera strategiska mål
- Skapa mätmetoder för uppföljning av dessa
- Att skapa modeller för att föra ut strategiska mål i verksamheten
- Att vara en länk mellan ledning och verksamhet

3.4.3 Anveckling, frekvens och typ

Bosse använder kalkylprogram varje dag och mer än halva sin arbetstid (ca 75%). Han började använda Lotus Symphony i slutet på 80-talet. Utbildningsbakgrund är bl.a. en kurs i makroprogrammering 1988. Efter denna kurs gjorde Bosse en del modeller genom att spela in kommandosekvenser, som sedan modifierades. Modifieringen av dessa makron blev dock komplicerade och tidsödande. Så småningom började Bosse fråga sig "Är jag programmerare eller ekonom?" De modeller som Bosse använder i sin verksamhet gör han till största delen själv. Han behöver mindre och mindre hjälp eftersom kalkylprogrammen blir mer och mer kraftfulla för varje version som kommer. De kalkylsystem som Bosse bygger är av alla typer: *Ordbehandlarsystemet*, *Lilla kalkylsystemet* och *Stora kalkylsystemet* (se kap 8.3 för klassificering av kalkylsystem). Han har även initierat och deltagit i utvecklandet av ett antal kalkylsystem av typen *Applikationen*.

Ordbehandlarsystem kan vara rena uppställningar av siffror. Små kalkylsystem kan innehålla beräkningar av mindre omfattande slag. Stora kalkylsystem är den vanligaste typen av Bosses kalkylsystem. Det är ofta dokument med flera flikar indelade efter produkter, råvaror, tidsperioder eller liknande. Dokumenten innehåller ofta många formler och är ofta länkade¹⁴ till andra dokument. Det finns mellan 30 och 50 grundmodeller som varieras på olika sätt med olika förutsättningar.

Bosse ser kalkylprogram framförallt som ett verktyg för kommunikation. Det innebär att han kan hämta och sprida information med hjälp av kalkylsystem av olika slag. Bosse har tillsammans med sin controllerkollega ansvar för underhåll av ekonomiaavdelningens modul i företagets rapportsystem, där man regelbundet sparar ner rapporter. Andra avdelningar har liknande system som förser företagets medarbetare med information om priser, förbrukning, produktion etc.

3.4.4 Anveckling, exempel och analys

Bosses anvecklingsverksamhet är omfattande och varierad. Gränsen mellan olika användningsområden kan formuleras på olika sätt. En indelning är efter storlek och komplexitet på kalkylsystemen. En annan indelning gäller *målgrupp*. Det påverkar utformningen om kalkylsystemet är till för eget bruk eller för någon annan. Målgruppens egenskaper kan också spela roll för utformningen. Om kalkylsystemet riktar sig till ledningen kan det innebära att urvalet av information skall ske på en högre nivå än om det riktar sig till produktionsavdelningen. En allt större del av Bosses tid åtgår för att, utifrån analyser av verksamheten och dess strategiska mål, utforma relevanta mätverktyg. Nedan redogörs för kalkylsystemens funktioner i termer av kommunikation, kontroll, formalisering och analys. Funktionskategorierna har genererats induktivt utifrån upprepade bearbetningar av intervjumaterial och studier av kalkylsystem.

¹⁴ Bosse försöker numera att undvika länkning om han misstänker att det kan medföra problem (se även kap 3.4.4).

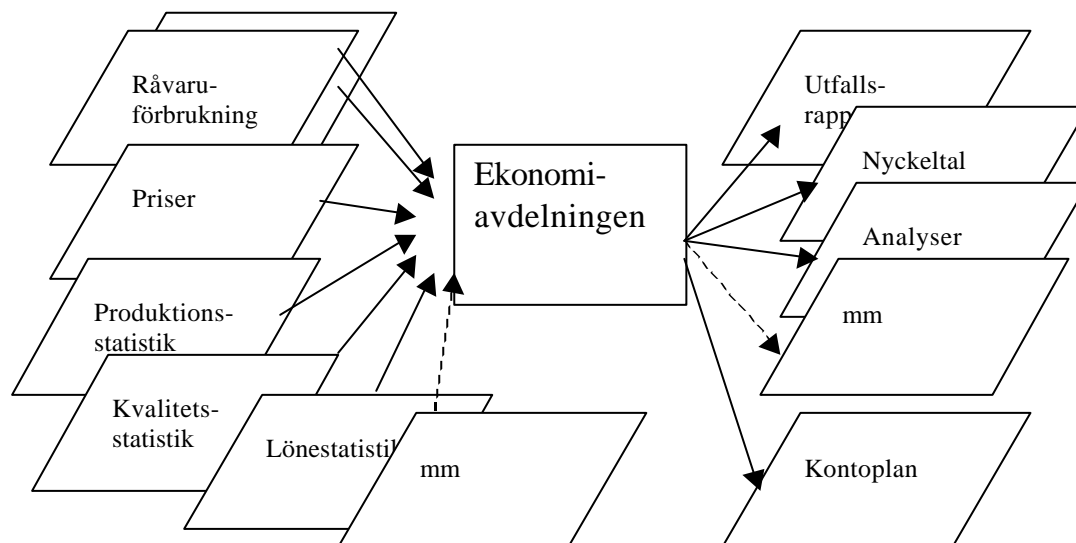
Kommunikationsfunktion

I och med att företagets lokala nätverk började användas har informationshanteringen på företaget ändrats i grunden. Där tidigare rapporter skickades med internposten, lagras nu digitala dokument på servrar. Bosse ser kalkylprogrammets viktigaste uppgift som möjliggörare av kommunikation.

Ett exempel på kommunikationsfunktionen är följande. Bosse får aktuella priser från inköpsavdelningen via ett kalkylsystem i det lokala nätverket där inköpschefen varje månad matar in de aktuella priserna på råvaror och produkter. Förbrukning av kemikalieråvaror lagras också i nätverket via ett annat kalkylsystem, som utvecklats på den avdelning, som har ansvar för uppföljning av kemikalieförbrukning. Förbrukning av massaråvaror registreras i ett annat kalkylsystem på samma sätt. Löneuppgifter uppdateras i ytterligare ett kalkylsystem. Kvalitetsdata för producerade produkter i ett kalkylsystem, produktionsuppgifter av olika slag lagras också via kalkylsystem för att bli tillgängliga på det lokala nätet.

För ekonomiavdelningen innebär detta att data för analys är tillgängliga för bearbetning via det lokala nätverket. De data som ligger lagrade i nätverket kan relateras till nya kalkylsystem, antingen genom program som hämtar data med automatik eller genom att användaren kopierar från dokument. Vidarebearbetning är sedan möjlig för den som önskar. Viss pappersdistribution har ersatts av digitala dokument. Den som vill kan givetvis skriva ut dokumenten på papper om det skulle vara ett önskvärt alternativ. Nätverket administreras av dataavdelningen som lägger upp kataloger och tilldelar rättigheter till olika användargrupper. Några av de system som ligger till grund för ekonomiavdelningens arbete används inte direkt av honom men har tillkommit på Bosses initiativ. Det kan ses som ett uttryck för *fadderanveckling* (se Tabell 13 sid 217).

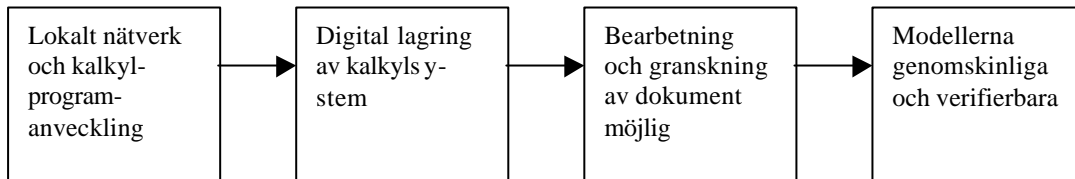
Bosse tillhandahåller i sin tur data för olika ändamål. Uppföljningsrapporter lagras på servern liksom aktuell kontoplan och olika nyckeltal. Nyckeltalen kan sedan användas



Figur 35 Ekonomiavdelningen och kalkylsystem i samverkan.

i andra kalkylsystem. Marknadsavdelningen kan t.ex. använda nyckeltal för produktionskostnadsberäkning för att beräkna priser till kund (se Figur 35 ovan).

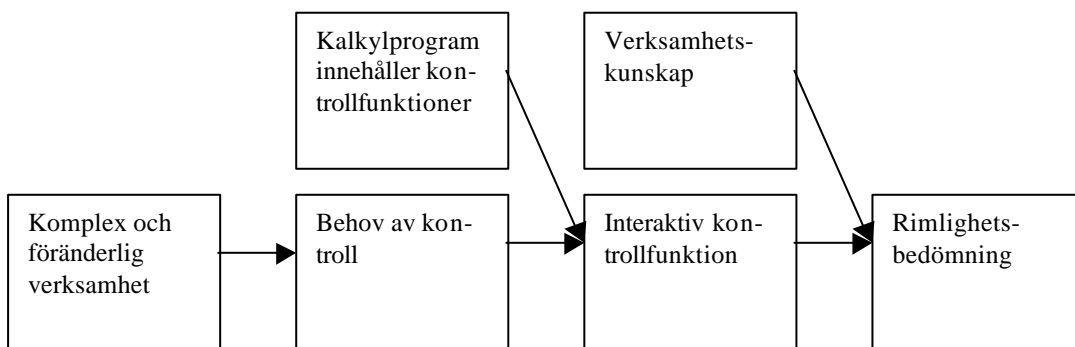
En effekt av att kalkylsystemen läggs ut på det lokala nätverket är att modellerna blir genomskinliga. Bosse ser kommunikation som ett sätt att överföra både information och kunskap om hur informationen kommit till. Ekonomiavdelningen och teknikavdelningen kan t.ex. byta modeller. "Då säger vi: 'Då kan Du se hur jag har räknat'. Folk vill ju hellre ha det på nätet än på papper. Då är det ju för att de kan se hur man har tänkt. Då slipper de ställa frågor som de ibland kanske upplever som genanta." (se Figur 36 nedan).



Figur 36 Effekter av lokalt nätverk och kalkylprogramanveckling.

Kontrollfunktion

Eftersom data används för att analysera verksamheten är det av helt avgörande vikt att den information som finns i rapporter av olika slag är korrekt. Då verksamheten är komplex och förutsättningar kan förändras, och då det inte är självklart hur olika värden och nyckeltal skall beräknas, så är gransknings- och kontrolluppgifterna som Bosse har centrala. Vissa standardrapporter som Bosse skapar görs i form av kalkylsystem. Ett exempel på detta, **produktkalkyl**, visas i Figur 38 nedan. Systemet vars syfte är att beräkna lönsamhet visar utfallsvärdena som jämförs med budgetdata. Indatavärdena kommer från andra system. Det kan vara långt från rådata till rapport och vikten av olika kontroller är stor (se ovan). Alla indata och utdata utsätts för en manuell *rimlighetsbedömning* (se Figur 37 nedan). Bosse kontrollerar regelmässigt såväl indata som



Figur 37 Förutsättningar för rimlighetsbedömningar

utdata genom att göra olika *kontrollberäkningar*. Dessa kontrollberäkningar kan göras i egna *permanenta* eller *tillfälliga kalkylsystem* eller direkt i dokumenten. Kalkylprogrammet tillåter att rapporter används för kontrollräkning när det är tillgängligt i digital form. Om t.ex. Bosse får en rapport om råvaruförbrukning kan kvoter (se Figur 128 sid 320) och nyckeltal i rapporten räknas ut direkt för en snabb rimlighetsbedömning. Om rapporten ligger i nätverket och är skrivskyddad kan beräkningarna ändå göras i

Industri

det skyddade dokumentet. Om det finns behov av att spara dokumentet, kan detta göras under annat namn på annan katalog. Denna *interaktiva kontrollfunktion* är alltid tillgänglig i ett kalkylprogram.

	A	B	C	D	E	F
1	PRODUKTKALKYL (kr/ton)					
2	1999 Mån tvk		Budget 99	Jan	Feb	Mar
3		Produktion ton	50 000			
4		Nettointäkt AB fabr	9 999	9 998	9 999	10 000
5		Reklamationer	-10	-10	-10	-10
6						
7	MÄNGDRÖRLIGA KOSTNADER					
8		Vedkostnad	-500	-500	-500	-500
9		Pumpmassa övriga mängdrörl	-100	-100	-100	-100
10		Blekt massa	-500	-500	-500	-500
11		Energi	-100	-100	-100	-100
12		Kemikalier	-20	-20	-20	-20
13		Emballage	-100	-100	-100	-100
14		S:a mängdrörl kostnader	-1 320	-1 320	-1 320	-1 320
15		Bidrag e mgdrörl. inkl rekl	8 669	8 668	8 669	8 670
16						
17	TIDRÖRLIGA KOSTNADER					
18		Pumpmassa,tidrörlig del	-500	-400	-400	-400
19		Övriga tidrörl.kostn.	-1 000	-1 000	-1 000	-1 000
20		Marknadskostnader	-200	-200	-200	-200
21		S:a tidrörl. kostn.(exkl kassarab)	-1 700	-1 600	-1 600	-1 600
22		Tot. tillv.kostn. inkl reklamationer	-3 030	-2 930	-2 930	-2 930
23		Kalkyldiff räntor	0	0	0	0
24		Fördeln diff	0	0	0	0
25						
26	RESULTAT		6 969	7 068	7 069	7 070

Figur 38 Del av produktkalkyl (OBS fiktiva värden).

Formaliseringsfunktion

De modeller som Bosse bygger innebär förenklingar. En ytterst komplex verksamhet som kan beskrivas på en oändlig massa sätt skall sammanfattas i några få mått. Bosses uppgift är att välja ut de intressantaste måtten.

"Jag måste göra förenklingar. Det är ju min uppgift i det här, översätta det här som alla vet och göra det begripligt och styrande. Det här är det som är min yrkeskunskap, att översätta det här. Produktionen, de kan ju sitta i detalj, mycket djupare än jag kan och ledningen de vet ju hur dom vill styra. Och sen sitter jag mitt emellan och skall välja ut de relevanta, viktigaste talen och göra det till nåt som man kan förstå. Om man har ett mål i form av tb/tim^{15} då skall man försöka översätta det till att reparatören som jobbar på elverkstaden, han skall veta, att om han reparerar den här grejen en timma långsammare så kostar det företaget 100.000 i timmen. Man ska kunna jobba från den här modellen. Det handlar om att man skall hitta ett företagsspråk som är enkelt. Som alla kan förstå. Sen är det chefernas sak att föra ner det här."

Ett kalkylsystem som Bosse jobbat mycket med är mätning av **täckningsbidrag per timme** (se Figur 39 nedan). Systemet syftar till att motsvara av ledningen formulerade

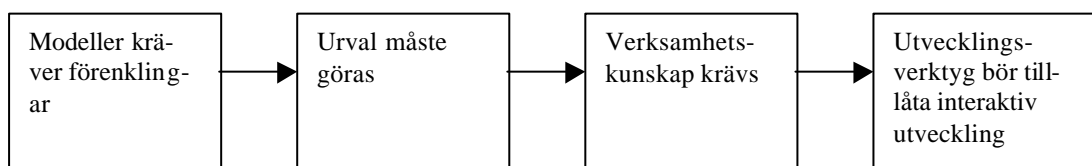
¹⁵ Täckningsbidrag/timme.

Användare och utvecklare

mål för hur verksamheten skall styras. Dessa mål kan sägas vara en effekt av analyser som utförts tidigare och som pekade på att dåvarande mätmetoder inte fångade upp alla önskvärda aspekter. Utarbetandet av systemet har inneburit en översyn av de data som finns i företagets system. Arbetet har lett till diskussioner kring olika mätvärden och hur de är konstruerade. Systemet kan sägas innebära resultatet av ett systematiskt ifrågasättande av hur verksamheten skall betraktas och mätas. Påverkande faktorer visas i Figur 40 nedan.

Lönsamhet beräknad på producerad kvantitet						Bidrag kr/tim						Verknings-				
Månad	Nom	Tot tid	Prima	Prislista	Styv-	Månadsutfall		Februari 1998		Prod.	Mängdörl	Verknings-	gradsdiff.	mängdörl	Bidrag	
						Bredd-	S:a pris	Summa	Intäkt spec.							(std 86%
2	ytvikt	tim	Ton	Ab-fabr	het	komp	Ab-fabr	intäkt prima	Prris Ab fabr	netto	Intäkt	verkn grad)	(14.-/%)	Kalkdiff	Mängdörl	Bidrag
	g/m ²	(e just f stopp)		kr/ton	kr/ton	kr/ton	kr/ton	kronor	kronor	ton/tim	kr/ton	kr/ton	kr/ton	kr/ton	kr/tim	kr/tim
Kvalitet 1																
	Prod 1															
	Prod 2															
	Prod 3															
	Summa															
Kvalitet 2																
	Prod 1															
	Prod 2															
	Prod 3															
	Summa															
	Totalt															

Figur 39 Förenklad beskrivning av del av kalkylsystem för beräkning av täckningsbidrag/timme.



Figur 40 Förutsättningar för formalisering.

Analysfunktion

Den funktion som Bosse poängterar mest och som han betraktar som mest prioriterad, och helst vill jobba med, är analysfunktionen. Till analysfunktionen hör att dels analysera befintliga data enligt etablerade modeller, dels att skapa verktyg/modeller för analys i enlighet med de strategiska mål som företagsledningen drar upp.

"Man vill ju gärna använda standardprogram, men man vill också kunna göra speciella analyser. Det är svårt att hitta standardprogram som täcker detta. Jag vill mycket hellre jobba med analysdelen. Jag vill hellre gå däråt, och då är kalkylprogram mitt redskap."

I Bosses yrkeskunskap ingår att han känner och förstår processen på företaget. Yrkeskunskapen omfattar både viss kunskap om produktionen och om ekonomiska modeller.

"Vi ser inte träden, vi ser skogen. Det är det som är effektivt ur företagets synvinkel även om vi och teknikerna i produktionen behöver varandra. Ekonomer utan tekniker vore väl lite blodfattigt kanske. Det är roligt att jobba med tekniker. Man ser olika på samma saker. Man drar olika slutsatser från samma data. Det är det som är fruktbart, roligt."

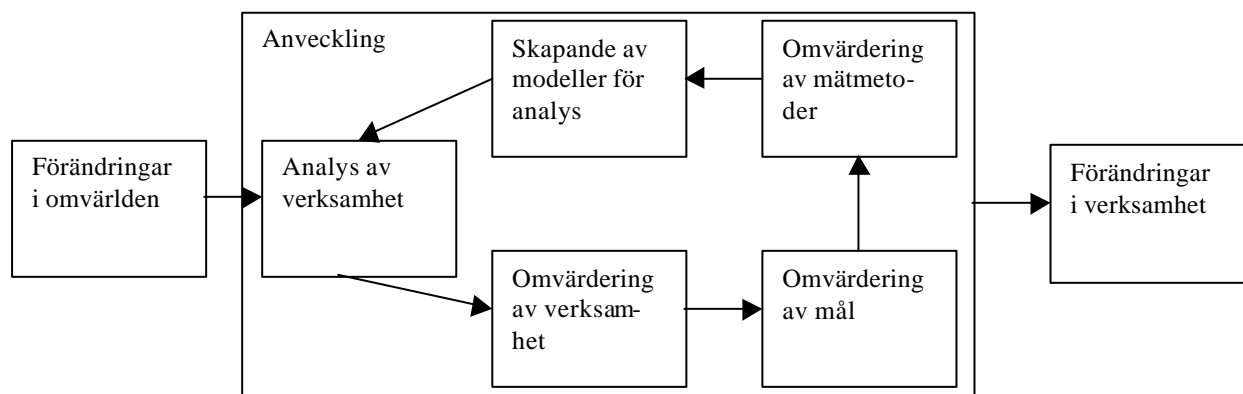
Varje månad gör man på ekonomiavdelningen en analys av resultat. Utifrån den ställs frågor som: Varför är oljeförbrukningen så här hög? Varför har elförbrukningen ändrats? De befintliga modellerna är instrument som säger att den förväntade resultatet vid viss produktionen skall vara ett visst. Då skall man bara kunna sätta in värden i modellen och se hur det blir, för att sedan kunna fråga varför blev det så som det blev?

Det försiggår ett ständigt omprövande av mätverktygen. Man strävar efter att utforma modeller som överensstämmer med de strategiska målen och som kan användas för kommunikation mot verksamheten. Då man skall hitta former för att styra mot strategiska mål innebär det fördelning av ansvar, diskussion om hur man skall göra modeller, vad modellerna skall fokusera, diskussion om vad är det som driver kostnaderna, vad är det som är rimligt och vad är det som påverkar vad.

Den önskade modellen skall beskriva verkligheten på ett sätt som gör att man kan ge ansvar längre ner så att det går att fastställa vem som har ansvar för vad och hur det skall följas upp. Traditionellt har produktionen ansvar för produktionen, men Bosse menar att det inte räcker. Det gäller att hitta relevanta mätmetoder för att kunna förstå varför hur resultatet blivit som det har blivit och sedan återföra detta till verksamheten.

"När man hittar redskapen för att analysera, då märker man hela tiden vad det är som är fel. På så sätt så förändras ju förutsättningarna. Och rätt som det är har man kommit till ett stadium, att nu kan vi inte bara köra det här med den organisation som vi har. Eftersom vi ser att det här inte är som vi trodde. Så då måste vi göra om organisationen. Det är ju så det blir hela tiden. Det glider hela tiden. Det nya är förändring. Det är det som är grejen. Man skall uppnå det optimala. Det hinner man aldrig till. Det är som att skjuta på rörligt mål hela tiden."

Analys av verksamhet med hjälp av modeller utformade i kalkylsystem kan innebära omvärdering av mätmetoder och utformande av nya sådana. Utformningen av nya mätmetoder kan tillsammans med förändringar i omvärlden innebära omvärdering av befintliga uppfattningar av verksamheten (se Figur 41 nedan).



Figur 41 Den kontinuerliga förändringscirkeln. Effekter av kontinuerliga förändringar.

3.4.5 Arbetssätt

Bosses anveckling utmärks av en blandning av användning och ett ständigt utvecklande av befintliga och nya modeller. Kalkylsystem blir oftast inte helt färdiga. Förfining av kalkylsystem fortsätter tills de inte används längre. Bosses programmerar inte utan använder de icke-procedurella funktioner som kalkylprogrammet tillhandahåller. Då applikationer behöver utformas med makroprogram anlitar Bosse denna kompetens på annat sätt. Arbetssättet kan sägas överensstämma med Avdic (1995a:180ff) då det innebär stegvis förfining, sammanflätade aktiviteter, direktmanipulerande arbetssätt och avsaknad av standardiserat arbetssätt (se även sid 189).

Kunskapen om kalkylprogram är spridd på företaget. På avdelningen är kunskapen utbredd. Samtliga datorer på företaget är utrustade med ett programpaket innehållande ett kalkylprogram. Kalkylsystem kan därför utbytas via e-post eller via gemensamma minnesutrymmet (intranät).

Förutsättningar för KPA:

- Kunskap om kalkylprogram är spridd på företaget och avdelningen
- Tillgång till kalkylprogram på företaget

De kalkylsystem som Bosse jobbar med är i allmänhet av typerna *Stora kalkylsystemet* eller *Applikationen*. Dessa kalkylsystem kan bli ostrukturerade. En problem som kan uppstå med sådana kalkylsystem är *ostrukturerade länkar*. Om dokument länkas samman finns flera risker. Dels finns risker att uppdatering inte sker om länkningen sker i flera led. Dels finns risk att strukturen blir ohanterlig och oöverskådlig, dels finns risk att dokument som är källdokument tas bort, vilket kan leda till felaktiga data. Ytterligare en risk med länkning är att länkarna kan bli statiska på så vis att verkligheten förändras så att länkningen inte stämmer. Bosse försöker numera att undvika att länka dokument om det finns risk för långa länkningskedjor eller ohanterliga strukturer. Statische länkar kan han ibland undvika med funktioner, t.ex. INDEX. Ett alternativ till långa länkningskedjor är att skriva ett makro, dvs att programmera, för som hämtar data för vidare bearbetning.

Vid utformning av kalkylsystem finns enligt Bosse en risk att ta med för mycket information. En uppgift som anvecklaren har är att sovra bland möjliga uppgifter. Den färdiga layouten bör samla utdata på en bildskärm som kan skrivas ut på en A4.

Tumregler för utformning:

- Undvik ostrukturerad länkning
- Hämta externa data med program
- Ta inte med för många variabler
- Utdata på ett A4

3.4.6 Sammanfattning - Bosse

Bosse ser kommunikationsfunktionen som viktigast för sin KPA. Anvecklingsaktiviteterna kan delas in i kontroll, formalisering och analys.

Kommunikation

Lokalt nätverk och kalkylprogramanveckling gör det möjligt att kommunicera med genomskinliga och verifierbara modeller.

Kontroll

Komplex och föränderlig verksamhet medför behov av kontroll. Kalkylprogram och verksamhetskunskap kan vara förutsättningar för interaktiv rimlighetsbedömning.

Formalisering

Utveckling av modeller bör ske interaktivt i en föränderlig verksamhet. Förutsättning för interaktiv modellutveckling är utvecklingsverktyg som tillåter detta och verksamhetskunskap.

Analys

Den kontinuerliga förändringscirkeln (se Figur 41 sid 79) innebär att en föränderlig omvärld skapar behov av kontinuerlig verksamhetsanalys som kan leda till omvärdering av verksamheten som kan leda till omvärdering av verksamhetens mål som kan leda till omvärdering av mätmetoder som kan leda till analysmetoder som kan leda till...

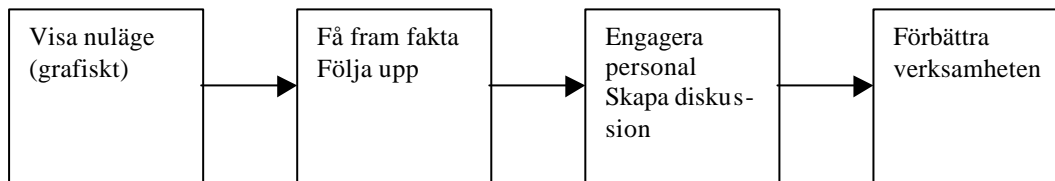
3.5 Thomas - avdelningschef

3.5.1 Uppgifter och verksamhetskunskap

Thomas har arbetat på företaget i tjugo år. Han är avdelningschef på efterbearbetningsavdelningen sedan ca fem år. Före det var han miljöansvarig. Efterbearbetningen omfattar omrullning, arkning och lager (se Figur 21 sid 57). Verksamheten bedrivs i kontinuerligt treskift. På avdelningen arbetar ca 70 personer.

3.5.2 Mål

Förutom de mål som beskrivits under Bosse och Ismo nämner Thomas särskilt att företaget har ett officiellt verksamhetsmål att bli en lärande organisation. Thomas menar att detta kan åstadkommas om det skapas diskussion om verksamheten bland personalen för att engagera den. Diskussion bland personalen kan åstadkommas genom att följa upp verksamheten och presentera fakta om verksamheten. Att engagera avdelningens personal kan åstadkommas genom att presentera verksamhetsuppföljningen grafiskt. Presentationen skall visa verksamhetsaspekter i form av utveckling och jämfört med budget. Det långsiktiga målet är att förbättra verksamheten (se Figur 42 nedan).

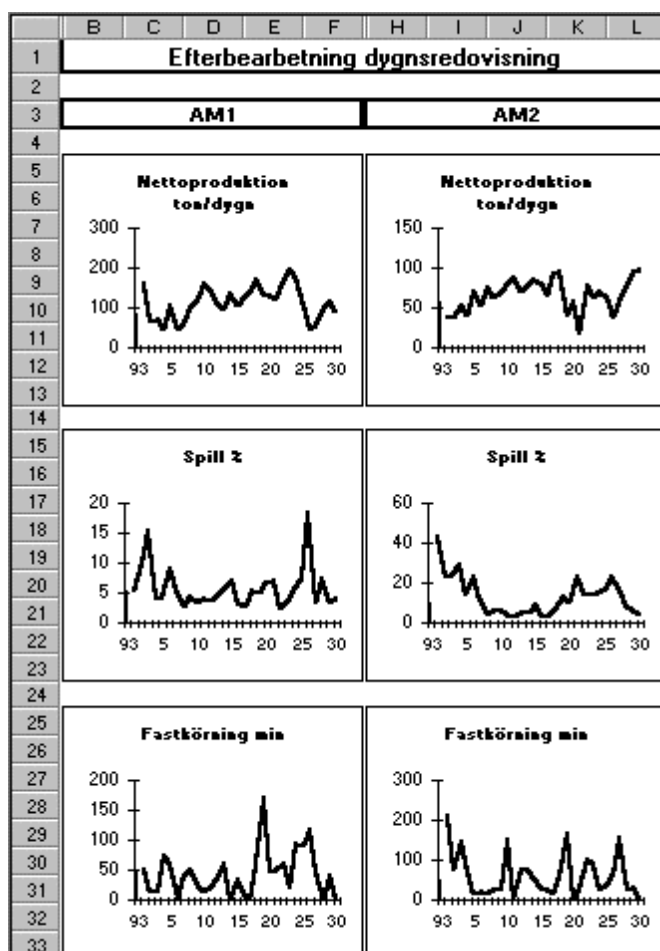


Figur 42 Mål efterbearbetningen.

Thomas menar att förbättring kan ske genom att vissa centrala mått skall följas upp. Exempel på sådana centrala mått är: spill, ton/tim, ton/dygn, ställtider och verkningsgrader. Det är viktigt att de som kan påverka dessa mått har det i sina tankar i sitt dagliga arbete. Det är inte bara chefer och arbetsledare som skall fundera på det. Uppföljning skall fokusera på att finna *avvikelser*.

3.5.3 Användning av kalkylprogram och presentationsprogram

Thomas är den av de tre respondenterna som själv använder kalkylprogram minst. Som utvecklare använder Thomas ett grafiskt presentationsprogram och i viss utsträckning kalkylprogram. Han har dessutom initierat utveckling av kalkylsystem av typen *Applikationen*.

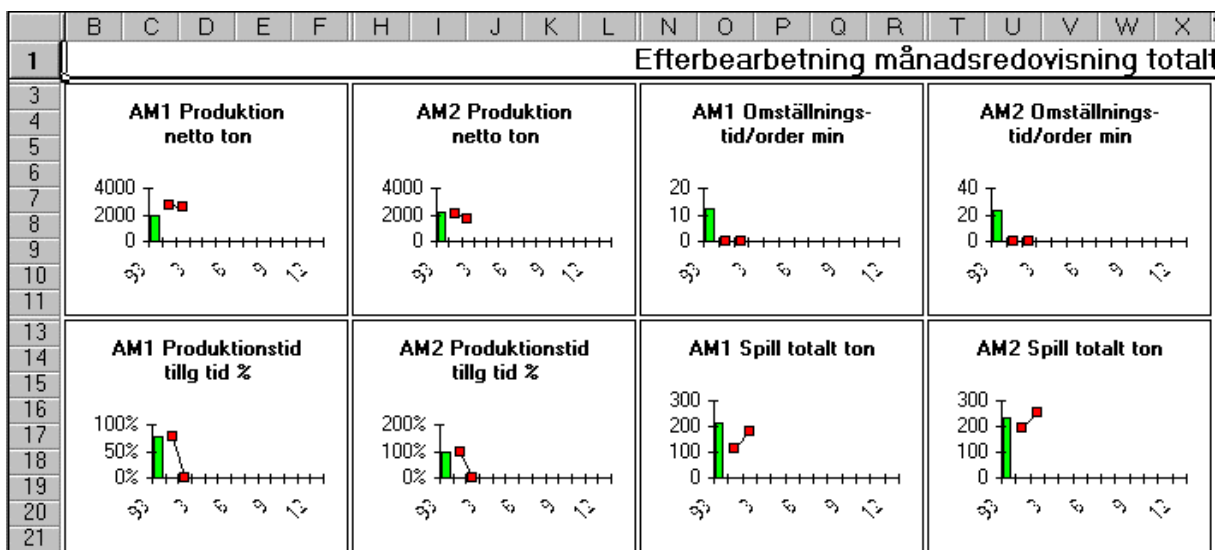


Figur 43 Del av kalkylsystem för uppföljning av produktionsdata dygnsvis.

3.5.4 Anveckling, exempel och analys

Thomas har i flera år arbetat med grafiska presentationer av verksamheten för att skapa intresse och engagemang. Genom att välja centrala verksamhetsaspekter och visa utveckling och jämförelse med budget har det på efterbearbetningen skett en utveckling mot mindre driftstörningar och därigenom högre verkningsgrad i produktionen.

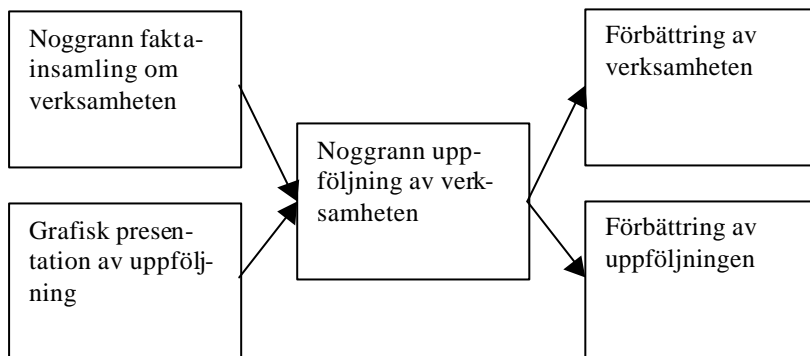
Thomas har initierat ett kalkylsystem för **uppföljning av arkningsmaskiner**. Systemet hämtar produktionsdata från ett centralt system och stopp och avbrottsdata från en intern databas som uppdateras av personal på avdelningen. Den lokala databasen kan användas för separata analyser men används också för att skapa rapporter dygnvis och månadsvis, absolut och ackumulerat. Rapporterna visar grafiskt (se Figur 43 ovan och Figur 44 nedan) och i tabellform utvecklingen av olika typer av stopp och avbrott.



Figur 44 Del av kalkylsystem för presentation av centrala variabler (OBS, fiktiva data).

Så småningom har produktionsledarna börjat att använda materialet vid presentation i sina grupper. Uppmärksammandet av stopp och avbrott har lett till att fastkörningar mer än halverats i timmar räknat på två år. Statistiken har också kunnat användas i förhandlingar med maskinleverantörer för att diskutera kvaliteten på maskinkonstruktionen.

Kontinuerlig och noggrann uppföljning av verksamheten ger möjligheter till förbättringar och nya uppföljningar. Om t.ex. en avbrottstyp blir omfattande kan denna analyseras särskilt och kanske delas upp på fler avbrottsorsaker. Analysen utmärks av en önskan att ta reda på fakta och att utgå från noggranna och korrekta uppgifter vid styrning av verksamheten. Den kontinuerliga uppföljningen och presentationen har skapat förväntningar från personalen. Man började efter en tid fråga efter diagrammen om de inte sattes upp på utsatt tid. En annan central egenskap hos Thomas presentationer är jämförelser. Utfall bör jämföras med något, budget, föregående år eller föregående månad. De som kan påverka resultatet måste märka när man gör bra ifrån sig (se Figur 45 nedan).



Figur 45 Faktainsamling och uppföljning av verksamheten.

3.5.5 Sammanfattning - Thomas

För att förbättra verksamheten strävar Thomas att förbättra uppföljning av densamma. Grunden för uppföljningen är noggrann faktainsamling och grafisk presentation av dessa. Faktainsamling och grafisk presentation kan utföras med kalkylprogram. Syftet är att åstadkomma diskussion och engagemang som i sin tur kan leda till förbättrad uppföljning.

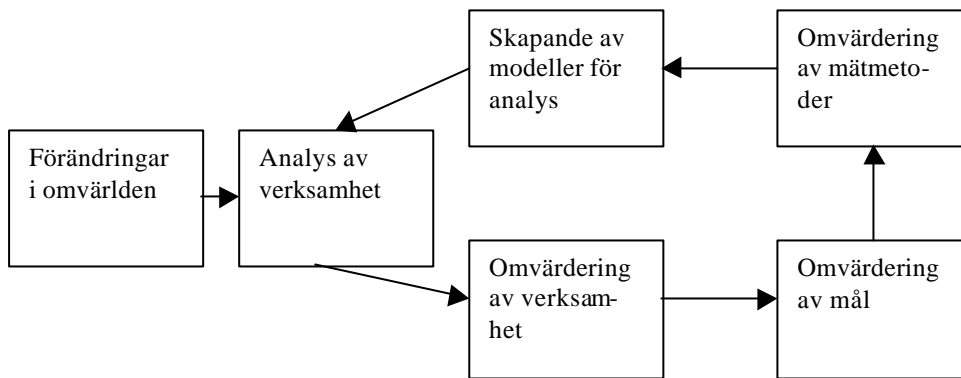
3.6 Sammanfattning - Industri

De tre respondenterna använder kalkylprogram på olika sätt men har flera gemensamma inslag i sin inställning till användning av kalkylprogram. Alla tre har en önskan om att utveckla verksamheten utifrån analyser baserade på fakta med hög noggrannhet och hög aktualitet. Inställningen hos respondenterna är påverkad av deras uppfattning om dels en föränderlig och konkurrensutsatt omvärld med många konkurrerande mål och dels den komplexa och kostnadskrävande verksamheten med höga krav på effektivitet och avkastning. För att kunna utföra flexibla analyser baserade på fakta och förståelse för förutsättningarna för verksamheten, krävs ändamålsenliga analysverktyg och verksamhetskunskaper som medför förmåga till rimlighetsbedömningar.

Den pågående omvärderingen av verksamheten och dess förutsättningar medför omvärdering av modeller och mätmetoder för värdering och uppföljning av verksamheten. Kalkylprogram kan stödja problemlösningsprocessen genom att modeller formaliserade i kalkylprogram blir kritiserbara och därmed möjliga att värdera. På så vis förs diskussionen upp på ett högre plan och principer för mätning och värdering fokuseras och inte bara värdena i sig.

Utveckling av IT i form av interaktiva verktyg för analys (t.ex. kalkylprogram), nätverk för distribution av information, och genomskinlig och manipulerbar information påverkar möjligheterna att analysera verksamheter. Den kontinuerliga förändringscirkeln i Figur 46 nedan åskådliggör ovan beskrivna förhållanden.

Industri



Figur 46 Den kontinuerliga förändringscirkeln.

Myndighet

"Vår verksamhetsidé 'Vi ska med kompetens, lyhördhet och personligt engagemang, erbjuda örebroarna stadsbyggnadsservice till efterfrågad kvalitet."

(Ur Årsredovisning 1997, Stadsbyggnadskontoret Örebro)

- *Detta kapitel syftar till att redovisa tre personers förutsättningar för och effekter av anvecklingsverksamhet, med speciell fokus på möjligheten att ta hänsyn till verksamhetskunskap och svårformaliserbara mål.*

4 Myndighet

4.1 Myndigheten

Myndigheten där studien gjorts är en kommunal myndighet, Stadsbyggnadskontoret i Örebro, med 95 anställda 1997-12-13. Myndigheten lyder under kommunstyrelsen och bereder ärenden för beslut i kommunstyrelse och byggnadsnämnd. Då besluten är fattade genomför myndigheten de fattade besluten. Myndigheten är uppdelad i avdelningarna Stab, Agenda 21, Stadsingenjörsavdelning, Plan och bygg, Trafik, Mark och exploatering, Bostadsanpassning, Park och Naturvård.

Som skattefinansierad kommunal myndighet är det ekonomiska målet att nå ett nollresultat, dvs inte gå med förlust och inte heller gå med vinst. I myndighetens årsredovisning formuleras verksamhetens mål (se *Bilaga 6*). I denna står bl.a. under rubriken *Vår verksamhetsidé* "Vi ska med kompetens, lyhördhet och personligt engagemang, erbjuda örebroarna stadsbyggnadsservice till efterfrågad kvalitet."

Myndighetens verksamhet berör i stor utsträckning tunga, långsiktiga frågor, som ofta berör och intresserar kommuninnevanorna. Det kan t.ex. beröra iordningställande av ett fritidsområde i anslutning till Hjälmaran, en sjö som ligger nära Örebro, men som aldrig kommit att användas som fritidsområde i någon större utsträckning av örebroarna. Myndigheten planerar också utformning och sträckning av olika vägar, en verksamhet som engagerar örebroarna och deras politiker i hög grad. Myndighetens handläggare syns inte sällan i tidningen då olika planer presenteras eller diskuteras.

4.2 Genomförande

För allmänna principer för metodanvändning hänvisas till kapitel 2. I detta avsnitt redovisas endast delprojektspecifika metodaspekter.

4.2.1 Metod

Undersökningen som genomförts, kan, liksom undersökningen på industrin, betecknas som en kvalitativ fallstudie genomförd med (1) öppna intervjuer och (2) i förekommande fall studier av kalkylsystem samt (3) informella samtal som genomförts vid varierande tillfällen.

Intervjuerna (1) har följt intervjumallen i *Bilaga 2*. De är inspelade på band och även nedskrivna manuellt vid intervjutillfällena. Registreringen har gått till så att anteckningarna först renskrivits med hjälp av ordbehandlare därefter har intervjuanteckningarna kompletterats utifrån bandinspelningarna. När registreringen varit klar har upprepade genomläsningar genomförts för att koda innehållet. Kodningen har gått till så att noteringar gjorts i dokumentet, men skilt från den löpande texten med intervjuregistreringen. Åtskillnad mellan texttyperna har gjorts med hjälp av ramar eller indragningar. Antalet intervjuer per respondent har varierat mellan två och tre. Respondenterna har haft tillfälle att kommentera och rätta de kodade intervjuerna.

Studier av kalkylsystem (2) har gjorts på kalkylsystem som kopierats vid intervjutillfället eller vid något annat tillfälle t.ex. vid ett samtal. Studierna har gått till så att systemen har bedömts och provats med avseende på deras relevans för hur respondenterna kunnat uppfylla svårformaliserade mål i sin verksamhet med hjälp av kalkylsystemet.

Registreringar av de informella samtalen (3) har främst gällt Olle och gjorts på det sätt som bedömts lämpligt vid eller efter samtalstillfället. Oftast har noteringar gjorts i anslutning till samtalstillfället på lappar som samlats för registrering, genomläsning och sammanställning på samma sätt som intervjuerna. Någon gång har noteringen gjorts vid ett helt annat tillfälle då något fenomen i omgivningen utlöste en association som resulterat i en reflexion med anknytning till respondentens verksamhet och kalkylprogramutveckling. Någon loggbok över hur många och hur långa samtal som förts har inte gjorts. En uppskattning är att antalet samtal, räknat över en femårsperiod kan vara ca 2 till 10 per respondent. Flertalet av samtalen har skett i samband med kursaktiviteter.

Ej refererade citat i texten är avskrifter från bandinspelningarna.

4.2.2 Urval av respondenter

Respondenterna har jag lärt känna vid en kurs i kalkylprogram som genomfördes på myndigheten. De har valts därför att jag, utifrån min kännedom om deras verksamhet och deras kompetens, bedömt att de varit representanter för en grupp personer som använt kalkylprogram för att uppfylla mål i verksamheten. Den första utvecklaren

Olle, är sysselsatt med myndighetens kärnverksamhet och använder kalkylprogram på flera olika sätt. Den andra anvecklaren, Jan, är sysselsatt med ekonomi och administration och använder kontinuerligt kalkylprogram som ett verktyg i detta arbete. Den tredje anvecklaren, Annette, har haft olika uppgifter i verksamheten och använt kalkylprogram vid enstaka tillfällen.

4.2.3 Disposition av presentation

Presentationerna av respondenterna och deras erfarenheter följer följande disposition:

- beskrivning av respondenternas arbetsuppgifter och verksamhetskunskap
- redogörelse för verksamhetsmål ur respondentens perspektiv och professionella mål
- beskrivande presentation av respondentens kalkylprogramanvändning
- beskrivning och analys av respondentens kalkylprogramanveckling i termer av förutsättningar och effekter
- beskrivning av respondentens arbetssätt och relaterade omständigheter
- sammanfattning

4.3 Olle - exploateringsingenjör

4.3.1 Uppgifter och verksamhetskunskap

Olle har arbetat på myndigheten sedan 1981 som exploateringsingenjör med fastighetsförvaltning och exploatering. Han är civilingenjör och har tidigare även arbetat som väg- och vattenbyggnadstekniker. Han sysslade då med VA-projektering, och har i det sammanhanget arbetat med analys av flödet i vattendrag, t.ex. Svartåns vatten. Även inom ramen för sina nuvarande uppgifter har hans specialkunskaper om vattenflöden kommit till nytta då han (med hjälp av kalkylprogram) beräknat effekter av vattenflöden vid konstruktion av t.ex. broar. Efter en omorganisation av myndigheten 1996 sysslar Olle med mark- och exploateringsfrågor. Före omorganisationen fanns även bygg- och fastighetsärenden bland den dåvarande avdelningens uppgifter.¹ Olle är enligt egen utsägo matematiskt lagd. ”Min personlighet gör att jag vill räkna på saker”. ”Jag har den läggningen att jag inte litar på intuition”. Bakgrunden som civilingenjör spelar roll tror Olle, bl.a. genom att presentationsaspekter inte alltid upplevs som så viktiga.

Olle har inom myndigheten ansvar för ett eget geografiskt distrikt för exploatering. Arbetet med exploatering innefattar aktiviteter från att köpa mark till att administrera utbyggnad av gator, ledningar och parker för nya områden, bilda tomter och sälja dessa. Före omorganisationen var hanteringen ”...från start till mål”. Exploateringen utgår från sk *råmark* dvs ”...mark med förväntningar...”. Arbetet innebär även upprättande av långsiktiga planer och innefattar beredskap för sysselsättningsfrämjande åtgärder.

¹ I kapitlet finns exempel på anveckling som Olle genomfört under hela sin tid på myndigheten.

Det finns även en avtalsrättslig sida där det ingår mycket avtalsskrivande. Arbetet kan t.ex. innebära träffande av samexploateringsavtal med större företag.

Inriktningen för avdelningen *Mark, skog och exploatering* där Olle nu arbetar formuleras enligt följande i årsredovisningen för 1997:

"Anskaffning och innehav av mark för tätbebyggelsens utvecklingsbehov samt friytor för rekreation, natur- och miljövård.

Förvaltning av markreserven i form av skogsbruk och utarrendering av jordar. Exploateringsverksamhet och avtal med enskilda fastighetsägare.

Anskaffning och förvaltning av vissa av kommunens bostäder för socialt udda boende inom utvecklingsområden. Förvaltning av tomträttsfastigheter."

Avdelningen där Olle arbetar kan betraktas som självständig, så till vida att kommunfullmäktige antar en långsiktig utvecklingsplan som sedan styr myndighetens arbete. Inom ramen för denna relativt allmänt hållna plan kan myndigheten agera självständigt.

4.3.2 Mål

Verksamhetsmål för Olles avdelning på myndigheten skulle kunna sammanfattas

- Långsiktig markförsäljning för kommuninnevärnans nytta, ekonomiskt, miljömässigt och socialt

- Rationell administration av kommunens fastigheter (gällde tidigare organisation)

Detta uppnås genom framtagande av relevant och korrekt beslutsunderlag till beslutsfattare i form av

- uppföljning av verksamhet
- likviditetsbedömningar
- fastighetsvärderingar
- nyckeltal för lönsamhetsbedömning

Specifika yrkesetiska mål som ingenjör och handläggare/administratör är

- Ändamålsenlighet
- Noggrannhet

De yrkesetiska målen underlättas av möjlighet till

- Översikt
- Bearbetningsbar information
- Flexibilitet
- Förenkling
- Utnyttjande av relevanta beräkningsmetoder

4.3.3 Anveckling, frekvens och typ

Olle började använda datorer i början av 1980-talet genom att privat köpa en Commodore 64. Till denna dator inköptes så småningom kalkylprogrammet *CalcResult*. Den första kalkyltillämpningen var administration av ekonomin på hyreshus. Olle satt även hemma och "...jobbade för jobbet", genom att göra årsvisa sammanställningar av exploateringsinvesteringar.

Olle säger sig inte göra så stora system. ”Jag använder mest för ekonomiska saker.” Arbetet går ut på att "...sammanställa och lämna rapporter för utfall och prognoser för exploateringsinvesteringar”. ”Jag hjälper till när economichefen frågar: Kan vi göra ett program för lönsamhet i investeringar?”

Ungefär halva sin arbetstid vid datorn använder Olle till att arbeta med kalkylprogram. Då Olle fått mer administrativa uppgifter har dessa tagit mer av hans totala arbetstid jämfört med när han enbart var handläggare. Flera kalkylsystem som Olle gjort, används av medarbetare på avdelningen, vilket gör Olle till en form av internkonsult. De system som Olle bygger är av alla typer, från *Ordbehandlarsystemet* till *Applikationen*.

4.3.4 Anveckling, exempel och analys

Olles beskrivning av effekter på arbetet som användning av kalkylprogram är: ”jag tycker att jag får beslutsunderlag”. En del av kalkylsystemen är till för att stödja kärnverksamheten och en del är till för att rapportera om verksamheten. ”Economichefen är väldigt nöjd med att vi kan lämna rapporter i denna form”.

Följande motiv ser Olle med användning av kalkylprogram:

- överskådlighet
- skapar diskussion - flexibelt
- snabbt att ändra parametrar ”våldigt snabbt!”
- snabb redigering (t.ex. kopiera samma värde)

Nedan redovisas exempel på kalkylsystem som Olle utvecklat/anvecklat.

Sammanställning

Översikter över objekt i exploateringsverksamheten kan utnyttjas dels som översikter (Ordbehandlarsystem) men även till beräkningar av olika slag. Nedanstående exempel (Figur 47 nedan) visar ett system för årlig **budgetering av exploateringsföretag**. Genom att summera radvis och kolumnvis ges en bild av likviditet.

Område	Mark	...	Gata	Park	Belysning	S:a utg.	S:a ink.
Bostäder norr							
Lanna							
Stäket kv.							
...							
S:a norr							
...							
S:a söder							
...							
S:a bostäder							
...							
S:a industri							
...							
S:a exploate- ring							

Figur 47 Struktur på exploateringssammanställning.

Användare och utvecklare

Från början sammanställdes alla objekt på samma blad, vilket gjorde hanteringen oöverskådlig. Så småningom har Olle delat upp objekten på flera blad enligt en logisk struktur, vilket har underlättat arbetet och skapat *översikt*. Jämfört med en manuell hantering var denna hantering en *förenkling*. Exploateringssammanställningen kunde också utnyttjas till simulering av olika utfall, vilket ger *flexibilitet* i planeringen. Exempel på en sådan fråga är: "Hur slår en minskning av parkinvesteringarna i ett visst objekt på hela likviditeten?" Ett makro finns utvecklat till exploateringssammanställningen som summerar värden för ett önskat år. Systemet, som används av ytterligare två personer, har varit i drift i ca tio år och har kommit att fungera som ett planeringsinstrument. Om man ser att kostnaderna för administration blir för stora kan planeringen revideras och projekt skjutas åt ena eller andra hållet i tiden.

För att förenkla administrationen av sin arbetstid och allokera tiden på olika projekt har Olle konstruerat en **tidrapport** (se Figur 48 nedan). Tidrapporten gör det möjligt att följa upp dels hur mycket tid som lagts ned på olika projekt och dels att följa upp hur arbete lagts ned på olika projekt över tiden. Det senare kan användas för att relatera nedlagd tid till beslutsprocessen för olika projekt. Om beslut t.ex. blivit återremitterade till myndigheten kan systemet visa vilka effekter detta fått på nedlagd tid. Systemet hjälper även Olle att hålla reda på semester och flextid. För att automatisera öppnandet av dokumentet så att Olle alltid fyller i när han börjar jobba, lagras dokumentet så att det öppnas då han startar datorn.² En fördel som Olle upplever med detta kalkylsystem (och andra) är att det går *snabbt att ändra parametrar* ”väldigt snabbt!” och att det också går *snabbt att redigera* (t.ex. kopiera samma värde).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	
1	TIDRAPPORT																											
2	Namn								År		Mån		Anst nr		Arbetsställe													
3																												
4	OLLE E								9 4		0 9		116		Mark och fastigheter													
5																												
6																												
7	Objekt	Konto					Summa	timmar																				
8	ÄGARKATEGORI anläggni	4	0	9	7	2	0																					
9	Servitut, Tomträtt	4	0	9	0	0	0																					
10	Budgetuppföljning	4	0	9	7	2	0																					
11	Budget 93	4	0	9	7	2	0																					
12	Försäljning, fastigheter	4	0	9	1	0	0																					
13	Tomträtt, friköp	4	0	2	0	8	0																					
14	V.bostr	4	0	9	7	2	6									4												
15	Köp, fastigheter	4	0	9	2	0	0																					
16	Örebro flygplats	4	0	8	3	3	8																					
17	Transportstaden	4	0	1	0	3	9																					
18	Brandstationstomten	4	0	9	2	0	6																					
19		4	0				0																					
20	Arla-Uno-X	4	0	9	7	2	0																					
21	Skebäcksleden/Grev Rose	4	0	0	4	2	2	19		1			4									1		2				
22		4	0				0																					
23	EXPLOATERING						0																					
24	Exploateringsadministratio	4	0	9	3	5	8									2							2					
25	Markens råd och indstent	4	0	2	1	0	2	4																				

Figur 48 System för tidrapportering.

² Om systemet lagras på katalogen XLSTART öppnas det automatiskt (gäller MS Excel).

Beräkning och simulering

Ett mindre system (se Figur 49 nedan) som Olle gjort används för att beräkna **alternativa hyreskostnader** för en investering vid variation i belopp, avskrivningstid, antagen inflation och internränta. Systemet är avsett att rationalisera beräkningsuppgifter och ge en säkerhet i uträkningen. Ett bredare beslutsunderlag erhålls då tre alternativ kan beräknas samtidigt. Systemet har ett gränssnitt med skydd på celler som ej skall ändras och formatering med färger och ramar för att styra inmatning och presentation.

5 KALKYLMETOD FÖR KAPITALTJÄNST, REAL ANNUITET						
6						
7	OBJEKT	ABC	OBJEKT NR			
8						
9						
10				FYLL I AKTUELLA UPPGIFTER!		
11				ALT 1	ALT 2	ALT 3
12	Köpeskilling och kostnader förenade med köp.		l kr	54 000	54 000	54 000
13	Avskrivningstid		A år	10	15	20
14	Antagen inflation		i %	0,00%	0,00%	0,00%
15	Den kommunala internräntan/lånräntan		p %	10,00%	10,00%	10,00%
16	REALRÄNTAN		z %	10,00%	10,00%	10,00%
17	Real annuitetsfaktor (betalning för 1 kr i slutet av varje period)			-0,163	-0,131	-0,117
18						
19	Kapitalkomponenten i första årshyran		K kr	8 788	7 100	6 343
20	som sedan alltså skall indexuppräknas för varje år					

Figur 49 Del av system för hyresberäkning vid investering.

För att beräkna framtida kapitalflöden har Olle gjort ett kalkylsystem för **beräkning av cash-flow³ och kapitalvärdesutveckling** för fastigheter (se Figur 50 nedan). Systemet är en investeringskalkyl för flera investeringar. Kan användas för såväl grundinvestering (fastighetsköp) som för kompletteringsinvestering (ombyggnad etc.). Systemet beräknar intäktseffekter och kostnadseffekter. Efter att alla parametrar fyllts i, visas kostnaderna med rött om affären är olämplig och med blått om den är acceptabel.

Systemet för cash-flow beräkning tillgodoser ett viktigt mål, nämligen att prognostisera om pengar måste lånas upp, eftersom systemet visar likviditeten vid olika tillfällen. Hyresutveckling och drifts- och underhållskostnader visas. Dessa värden ger möjlighet att beräkna det centrala nyckeltalet *driftnetto*⁴. Systemet visar vilka effekter en tilläggsinvestering ger, vilket kan skapa underlag för förhandling med hyresgäst som vill ha något. Systemet är till för att skapa ett beslutsunderlag vid investeringsbedömning. I och med att systemet ger ett bättre beslutsunderlag tillgodoses verksamhetens mål som refererades i början av kapitlet.

³ Betalningsströmmar.

⁴ Driftnetto = hyra - (drift + underhåll).

Användare och utvecklare

LÖNSAMHETSKALKYL FÖR INVESTERINGAR																					
BETALNINGSTRÖMMAR (CASHFLOW-ANALYS)																					
KOSTNADER UNDER EN TJUOGÅRS PERIOD I NOMINELLA BELÖPP											ANVISNINGAR										
Bedöm de kanakostnader som investeringar.											Endast guld rutor ifylls i resp. år penningvärde										
UTFALLET UTTRYCKT SOM NUVÄRDET AV SAMTLIGA KOSTNADER O INKOMST.F.R											Blått är viktigare resultat										
OBJEKT: ABC											Endast helt eller delvis värden som indata kan angras exakt										
											Se för örigt blad GN-TS förklaringar										
FÖRUTSÄTTNINGAR		GULDRUTOR IFYLLES																			
Kalkylränta (nominell)		10,50% (procent)																			
Inflation, kartkostnadsutveckling		3,50% (procent)																			
Hyresindexinflationen, faktor		0,70 sam. ger 2,45% hyresutveckling																			
Grundinvestering år 0		54 (kkr)																			
Investeringar restvärde år 20		10 (kkr)																			
		övriga investeringar och restvärden ifylls direkt i rchomet enligt användning av värde minskning beräkning																			
ÅRSVISA BETALNINGAR		Betalingar räknar vid slutet av varje period																			
UTBETALNINGAR, KOSTHÖKNINGAR:		Nuvärdet beräknar från slutet av perioden till slutet av år null																			
INBETALNINGAR, BESPARINGAR:		Negativa belopp																			
		Positiva belopp																			
BETALNINGAR MÅR		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20																			
INVESTERINGAR																					
GRUNDINVESTERING I1 kkr		-54																			
RESTVÄRDE ÅR 20 R1 kkr		10																			
NUVÄRDE AV R1		1																			
ÖVRIGA INVESTERINGAR I2 kkr		0																			
NUVÄRDE AV ÖVRIGA INVEST.F.R		0																			
SUMMA NUVÄRDE AV ÖVR. INV.F.R		0																			
RESTVÄRDE ÅR 20 AV ÖVR. INV.F.R		0																			
SUMMA RESTVÄRDEN R2 AV I2		0																			
NUVÄRDE AV R2		0																			
INTEKTS EFFEKT																					
HYRESÖKNING FÖR STÅRETT AV RESP. INVESTERING (börjar året efter)		0																			
SUMMA HYRESÖKNINGAR I PERIODEN		0																			
NUVÄRDE AV HYRESÖKNINGAR		0																			
RÄNTEBIDRAG FLERBOSTHUS 15-årsperiod																					
GÄRNTFRÄNTANS STORLEK x		9,00%																			
av investeringar beloppet kkr		54																			
BIDRAGSPROCENT av garantiränta		0,00%																			
ger RÄNTEBIDRAGET kkr		0																			
NUVÄRDE AV RÄNTEBIDRAG		0																			
SUMMA NUVÄRDE AV RÄNTEBIDRAG		0																			
KOSTNADEFFEKT																					
KOSTNADSBESPARING FÖR STÅRETT AV RESP. INVEST. (börjar året efter)		0																			
SUMMA KOSTNADSBESPARINGAR I PERIODEN		0																			
NUVÄRDE AV KOSTNADSBESPARINGAR		0																			
		1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869	1,67535	1,73399	1,79468	1,85749	1,9225
			1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869	1,67535	1,73399	1,79468	1,85749
				1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869	1,67535	1,73399	1,79468
					1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869	1,67535	1,73399
						1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869	1,67535
							1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396	1,61869
								1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107	1,56396
									1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997	1,51107
										1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106	1,45997
											1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629	1,4106
												1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681	1,3629
													1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228	1,31681
														1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926	1,27228
															1	1,035	1,07123	1,10872	1,14752	1,18769	1,22926
																1	1,035	1,07123	1,10872	1,14752	1,18769
																	1	1,035	1,07123	1,10872	1,14752
																		1	1,035	1,07123	1,10872
																			1	1,035	1,07123
																				1	1,035

Figur 50 Del av system för cash-flow beräkning.

Ett annat kalkylsystem som Olle byggt är ett **tomtpriserberäkningssystem**, som byggt för att fastställa hur man skall prissätta tomter (se Figur 51 nedan). Ett nytt system skapades för att anpassa den tidigare reglerade hanteringen till ett marknadstänkande. Systemet skulle spegla folks benägenhet att betala för tomter. Man fick konstruera kriterier utifrån detta med plus och minuspoäng (t.ex. för trafikbuller).

Huvudfaktorer för att beräkna pris är läge och byggbarhet.

Faktorn *läge* omfattar kriterierna

- orientering väderstreck
- attraktionskraft t.ex. område med hög status
- natur park på eller bredvid tomt
- geografiskt relaterad till attraktionskraft

För varje kriterium ges poäng mellan 1 och 3 där 1 är bäst och 3 är sämst. Som bäst kan en tomt få fyra poäng (4*1) och som sämst 12 (4*3). Vid fyra poäng multipliceras priset med 1,5.

Faktorn *byggbarhet* omfattar kriterierna

- form skall ej vara spetsig
- topografi slät = 1
- grund på grund ger max 7 poäng

Även här ges poäng som multipliceras med priset.

SMÅHUSTOMTER PRISSYSTEM 92							
Kalkylblad för många tomter							
Bedömningshjälp							
LÄGE orientering, attraktivitet enligt zonkarta, naturvärden, centralt/ytterkommunen							
	LÄGE				SUMMA		
KLASS	Orient	Attrak	Natu	Geogr	Totalt	poäng	faktor
1-3						4	1,50
1(0)=bäst						5	1,40
3=sämst				FAKTOR	###	6	1,30
tomt invid matarg ger 1 extra						7	1,20
entrég i söder ger 2						8	1,10
nyexpl på åker ger			3			9	1,00
gammal trädg mm ger			1			10	0,90
						11	0,80
						12	0,70
BYGGBARHET tomtform, topografi, grundläggning							
	BYGGBARHET				SUMMA		
KLASS	Form	Topo	Grundl			poäng	faktor
1-3(-7grund)				TOTALT		3	1,00
				FAKTOR	###	4	0,95
1=bäst						5	0,90
3 (7)=sämst						6	0,85
uppfyllnad ger<=			3p	grundl		7	0,80
pålning ger <=			4-7p	"		8	0,75
kort skaft ger <= 2p						9	0,70
långt tomtskaft ger <=5p						10	0,70
						11	0,70
VÄRDETIDPUNKT	10-95		GRUNDPRIS kr		106 836		
INDEX E84 fr 93-07-0	1,07		AREAPRIS kr/m ²		96		

Figur 51 Del av system för värdering av tomtpriser.

Beräkningen är även indexjusterad mot det s.k. entreprenadindexet för vägar (E84), vilket gör att modellen följer de kostnader som kommunen faktiskt har för sin verksamhet. Då man tidigare ej haft manuell prissättning, fick modellen utformas samtidigt med att kriterierna i den diskuterades fram. En marknadsorienterad nyordning i kommunen skapade behovet. Och utformningen på kalkylblad fungerade som ett medel att kartlägga hur värderingen egentligen skulle gå till. Kalkylsystemet bidrog till att belysa och skapa diskussion kring frågan. När modellen var klar beslutades faktorerna av kommunfullmäktige. Den utarbetade kalkylmodellen var en del av beslutsunderlaget.

Modellen fanns ursprungligen på ett kalkylblad. Då en ny version av kalkylprogrammet medgav, bröts systemet ner på flera blad för att ge en mer logisk struktur. Nyckeltal placerades på ett blad. Beräkning av en tomt och beräkning av flera tomter placerades på olika blad. Uppdelning gav översikt men behöll logiska relationer med länkar. Systemet kan användas för att beräkna priser på flera (upp till 80) fastigheter. Då samlas uppgifterna för alla fastigheter på ett blad. Bladet, som innehåller föreslagna priser och de beräkningar som lett fram till priserna, kan skrivas ut och fungera som ett beslutsunderlag vid diskussioner om det skäligen i prissättningen. Under tiden som systemet använts har modellen förändrats en del. En förändring är att vikten för olika

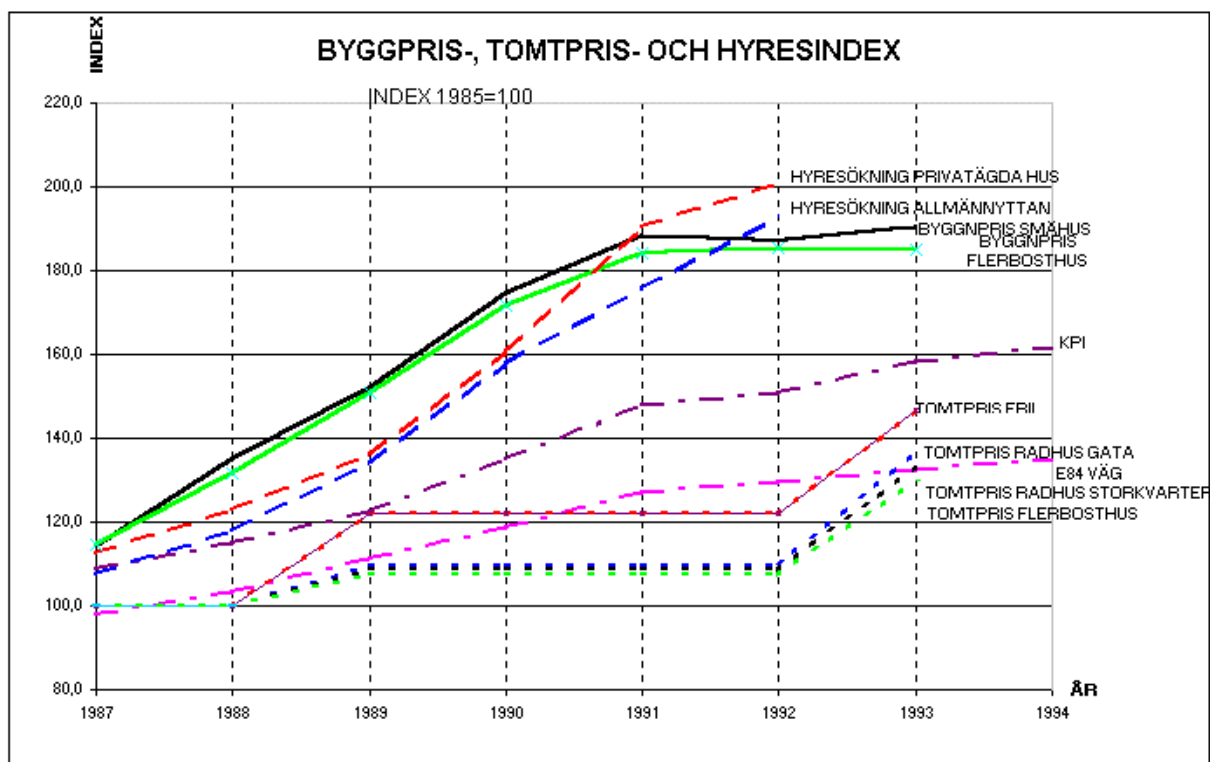
poäng har justerats i enlighet med den kunskap man har på avdelningen om hur fastigheter värderas. Denna förändring har varit ytterst enkel då det bara varit att ändra värdet i en cell på kalkylbladet. En annan förändring är att tomter över 700 m² fått justering på värdet av arealen som överstiger 700 m². För att få modellen att hantera detta har Olle modifierat kalkylsystemet. I det nya beslutsunderlaget framgår numer även de olika priserna på mark under och över 700 m².

Systemet kan relateras till flera av målen med den kommunala verksamheten. Dels kan priserna relateras till kommunens kostnader, dels konstrueras ett pris som skall motsvara ett marknadsvärde, vilket är en rättviseaspekt mot kommuninnevanorna.

Grafisk presentation

Gränsen mellan grafisk och annan presentation är flytande i kalkylprogrammiljö, dock kan de system där diagram förekommer sägas utgöra en egen aspekt av grafisk presentation.

Ett kalkylsystem, som Olle gjort, visar **pris- och indexutveckling**. Systemet innehåller "...en faslig massa siffror", vilka presenterar årsvis utveckling av ett antal nyckeltal för byggpriser, tomtpriiser och hyror (se Figur 52 nedan). Bland kurvorna finns konsumentprisindex (KPI) som en jämförelsetrend. KPI använder Olle i flera system och har därför skapat ett eget blad som kan utnyttjas i flera applikationer. Ett annat jämförelsetal är E84, som visar kostnadsutvecklingen för kommunens entreprenörer. E84 är



Figur 52 Del av system för jämförelse av pris- och indexutveckling.

uppdelat på vägbyggnation och husbyggnation. Systemet ger underlag för beslut om prissättning i olika avseenden. Jämförelser med E84 kan användas för att reglera kost-

nader så att dessa ej blir oskäligen åt något håll. Den kommunala målsättningen med nollresultat är viktig för myndigheten. Olle brukar ta med sig en OH-bild av trenderna på olika möten om han känner på sig att diskussionen kan komma att komma upp. En förtjänst med systemet är att det ger *möjlighet till sammanställning och översikt*. Enligt Olle kan det dock vara lite väl många linjer på diagrammet. De kan vara svåra att skilja från varann, speciellt då man inte har tillgång till färgskrivare. En egenhet med systemet är att det inte innehåller några beräkningar, endast tabeller med värden och en grafisk presentation av dessa värden.

Ett annat exempel på grafisk presentation gäller **presentation av förbrukning**. Olle har gjort ett system som visar månadsvis förbrukning av el och olja i hyresfastigheter. Med hjälp av kurvorna kan läckage eller andra avvikelser upptäckas lättare än om ingen sammanställning gjordes. Speciellt belysande blir sammanställningen då den kan göras över tiden och då jämförelser kan göras från år till år. Uppföljning av reparationer och investeringar underlättas av uppföljningen. Beslut för planering av nya investeringar kan också fattas med mer komplett underlag om effekter av tidigare investeringar kan infogas i underlaget.

Arbetsuppgiftsrelatering

Visserligen är samtliga Olles kalkylsystem exempel på utförande av arbetsuppgifter, men nedan visas några system som är intimt relaterade till Olles arbete som exploateringsingenjör.

Bland Olles kalkylsystem finns ett antal, som fungerar som register över viss typ av objekt, t.ex. över kommunens tomträttsfastigheter. I anslutning till denna tabell finns ett system för **tomtprisberäkning**. Med hjälp av systemet kan Olle jämföra nuvärdet av framtida inbetalningar med det kommunen skall sälja för, för att det skall gå jämt upp. Systemet kan också användas till att beräkna vad man kommer att tjäna på eller förlora på tomträttsverksamheten. Ett flertal variabla faktorer används för att utföra beräkningen t.ex. kalkylränta och inflation. Då kommunen skulle sälja tomträttsfastigheter till det kommunala bostadsbolaget, gjorde Olle förslag på rimligt friköpsbelopp. För att utföra beräkningen tog Olle först in externa data för att utifrån dessa beräkna lämpligt belopp. Bakom systemet finns en historia om hur kommunerna på 1940- och 1950-talet tog beslut om att upplåta tomträtter för egnahemsbyggande. Sedan dess har olika uppfattningar styrt synen på värdering av tomträtter. Systemet kan ses som en formalisering av den rådande synen på värdering av tomträtter. Idag innebär denna syn att kommunen skall få kompensation för värdeökning enligt KPI men inte mer.

Ett annat system med koppling till tomträttsystemet är ett system för **administration av tomträttskontrakt**, som Olle också gjort. Systemet hämtar aktuella uppgifter om tomträtter från centrala fastighetsregistret i Gävle, sammanställer och formaterar dessa uppgifter för att sedan koppla till ordbehandlingsdokument, som kan skrivas ut på arkivbeständigt papper och skickas till tomträttsinnehavarna. Systemet producerar även rapporter som kan tillställas politikerna för information om vad som förnyats och till vilket pris. Systemet integrerar interna och externa data, utnyttjar inspelade makron, integrerar programmiljöer och distribuerar information. Systemet verifieras av Olle

med en uppsättning poster som används speciellt för detta ändamål. Då Olle utformat systemet har han blandat procedurrell makroinspelning och deklarativ formelformulering. Vissa kommandosekvenser, t.ex. infogning av blad, kopiering av data och infogning av rader och kolumner, har spelats in. Formler har lagrats i ett speciellt blad och kopieras där av makrot för att klistras in i målbladet. En speciell formel omvandlar numeriska värden till värden skrivna med bokstäver för inmatning i kontraktet som skickas till tomrättsinnehavarna. Skälet till att denna omvandling görs är att kontraktbeloppet måste finnas både med siffror och bokstäver och att om det ena ändras måste det andra ändras för att undvika komplikationer.

Ett kalkylsystem fick byggas då en del av kommunen skulle bryta sig ur och bli egen kommun gjorde Olle ett system för **markvärdering** för att värdera kommunens mark så att skulder och tillgångar kunde fördelas rättvist. Systemet har också använts inför bokslutet för 1997 som innehöll nya anvisningar för redovisning av mark- och fastighetspriser.

Som tidigare nämnts är Olle expert på beräkning av vattenflöden. Denna kunskap har han använt till att konstruera ett kalkylsystem för **dimensionering av bropelare** på en bro som projekteras av myndigheten. Systemet formaliserar och tillämpar befintliga kvantitativa teorier om effekter av vattenflöden. Systemet bygger förutom på beräkningsformler även på historik om vattenflöden. För att göra sig en bild av vad som händer vid olika vattenstånd och olika flöden kan kalkylprogram användas för att interpolera och extrapolera. Genom att bearbeta statistik över vattenflöden kunde Olle få ett statistisk mått på hur ofta olika typer av vattenflöden inträffade. Detta kunde sedan användas i underlaget, då investeringsbesluten fattades.

4.3.5 Arbetssätt

Nackdelar

Nackdelar som Olle noterar med att utveckla kalkylsystem är:

- Stora kalkylsystem kan bli oöverskådliga
- Risk att förstöra formler om man av misstag matar in värden i celler med formler
- Risk att förändra formler då man förändrar kalkylsystemet
- Kan vara svårt att referera rätt om kalkylsystemet är stort
- Om kalkylsystemet endast skall utföra någon enstaka beräkning kan det ta onödigt lång tid
- Svårighet att nå data som ligger i databaser i eller utanför organisationen
- Utskriftsproblem
- Stabilitet på det lokala nätet
- Programstopp vid öppning av länkade blad vid makrokörning
- Versionsbyten

Olles lösningar på de fyra första problemen ovan

- Uppdelning på flera blad
- Att dela skärmen och låsa rubriker

Myndighet

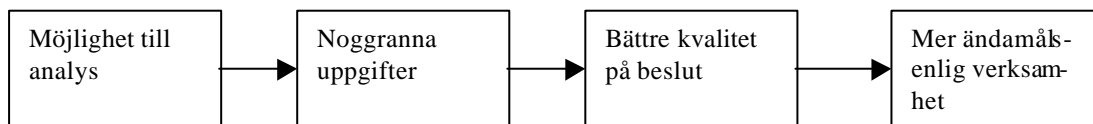
- Att strukturera utformningen av kalkylbladet och ge olika typer av områden olika färger. Olle formaterar ändringsbara celler med ljus bakgrund och med röd text för att hitta dem
- Användning av kommentarer och ledtexter
- Skydd av celler med data som inte skall ändras
- Strukturera systemet logiskt så att sammanhörande data placeras tillsammans

Makro kan användas när andra funktioner saknas. Olle har gjort makron för samma n-ställning av data och för att positionera sig på ett visst kalkylblad.

Eftersom Olle har lång erfarenhet av KPA och goda kunskaper om kalkylprogram får han ibland skapa kalkylsystem till andra på avdelningen. Att göra system, då behovet finns hos någon annan, uppfattar Olle som svårare än att göra åt sig själv. Skälet är att om man själv har ansvar och kunskaper så vet man hur det skall vara. Om man har ansvar för verksamheten har man ansvar för kalkylsystemet.

4.3.6 Sammanfattning - Olle

Olle är sysselsatt med myndighetens kärnverksamhet och använder kalkylprogram på flera olika sätt, för att administrera sin egen verksamhet (t.ex. tidsredovisning), för att rationalisera arbetsuppgifter (t.ex. cash-flow-beräkning), för att skapa nya analys-hjälpmiddel (t.ex. tomtrissystem). Olle tror att hans bakgrund som ingenjör spelar roll för användningen av kalkylprogram. En önskan att analysera, kartlägga och ta fram *relevanta, noggranna uppgifter* utmärker Olles verksamhet. Skälet är att ju noggrannare underlag, ju bättre kvalitet på beslut och ju mer ändamålsenlig verksamhet (se Figur 53 nedan).



Figur 53 Effekter av möjlighet till noggrant beslutsunderlag.

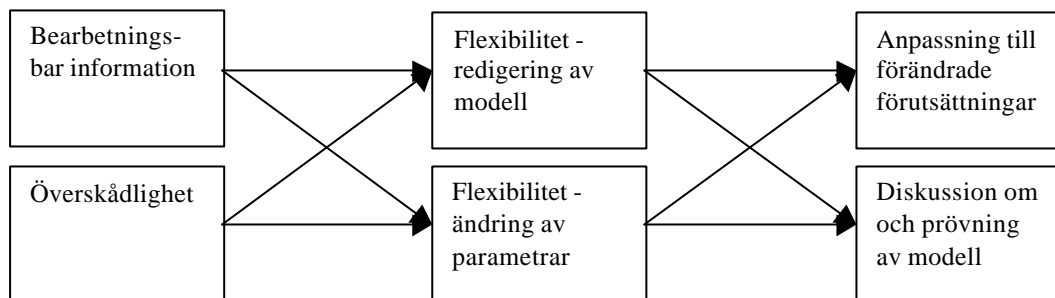
Verksamheten är regelstyrd. Politiska beslut ger de yttre ramarna, och därmed handlingsnormer för verksamheten. De centrala systemen kan inte återspegla alla förändringar i dessa beslut lika snabbt som de fattas. För att skapa kalkylsystemen i Olles verksamhet krävdes djup kunskap om lagar och bestämmelser, ekonomi, exploatering och byggnation. Systemen är i själva verket en formalisering av kunskaper om verksamheten.

Egenskaper som Olle relaterar till anveckling med kalkylprogram är:

- överskådlighet
- flexibilitet - ändring av parametrar
- flexibilitet - redigering av modell

Egenskaperna ger möjligheter till analys av befintliga uppgifter och även möjligheter att ompröva och skapa modeller för analys. Egenskaperna kan skapa en flexibilitet som kan ge ett mer omfattande beslutsunderlag än vad annars skulle vara fallet. Möj-

ligheten att redigera gör det möjligt att ändra förutsättningarna för beräkningar och analyser, vilket kan vara till nytta om verksamhetens förutsättningar förändras, t.ex. i form av nya politiska beslut (se Figur 54 nedan).



Figur 54 Effekter av överskådlighet och bearbetningsbar information.

Redigeringsfunktionen i kalkylprogrammiljö är i grund och botten ett verktyg för systemutveckling. Med redigering kan den grundläggande modellen förändras.

Ansvar finns för att kalkylsystemen ger rätt utinformation om ansvar finns för verksamhet. Systemen är verktyg i Olles arbete. Det är för Olles del inte systemen som producerar information utan Olle själv. All utinformation utsetts för rimlighetsbedömningar.

4.4 Jan - kamrer och stabschef

4.4.1 Uppgifter och verksamhetskunskap

Jan har arbetat i åtta år på myndigheten. Han är kamrer och tillförordnad stabschef. Han är utbildad ekonom och arbetar inom ramen för sina kamrersuppgifter med ekonomistyrning, budgetering, budgetuppföljning, redovisning och bokslut. Det ekonomi-relaterade arbetet sönderfaller i :

1. Ekonomistyrning (att påverka framtiden)
2. Ekonomi administration (dokumentation av ekonomiska händelser)

Stabschefsuppgifterna innebär uppgifter som arbetsledare för staben, inklusive avdelningarna information, personal och ekonomi.

4.4.2 Mål

Den i årsredovisningen deklarerade målsättningen (se *Bilaga 6*) är det övergripande styrinstrumentet för myndigheten, och därmed också staben som Jan tillhör.

För kommunekonomer finns också tretton etiska regler formulerade (se *Bilaga 6*). Reglerna beskriver ett förhållningssätt gentemot kommunal ekonomi och är formulerade av Föreningen Sveriges Kommunalekonomer. Bland reglerna märks t.ex. följande punkter:

Myndighet

"1. Visa respekt för och uppträd så Du stärker förtroendet för kommunal verksamhet

2. Verka för god ekonomisk hushållning och för att ekonomiska aspekter beaktas i verksamheten

6. Dölj aldrig relevanta fakta om kommunens ekonomi och ta de initiativ som krävs för att bevara en god ekonomisk hushållning

7. Verka för att beslut fattas efter saklig beredning och i medvetande om de ekonomiska konsekvenserna

8. Var opartisk till alla kommunmedborgare och alla förtroendemän i kommunen

9. Värna om öppen och korrekt information

11. Tänk på att Du förvaltar allmänna medel. Spekulera inte, ta inga risker."

(Etik för kommunalekonomer. Tretton etiska regler inom Föreningen Sveriges Kommunalekonomer)

Jan kan, utifrån sina arbetsuppgifter, sägas ha följande mål med sitt arbete:

- att biträda myndigheten, kommunstyrelsen och kommuninnevånarna att använda pengarna på mest effektiva sätt
- att bedriva ekonomifunktionens verksamhet på ett resurseffektivt sätt
- att bedriva ekonomifunktionens verksamhet på ett sakligt, öppet och korrekt sätt

4.4.3 Anveckling, frekvens och typ

Jan har använt kalkylprogram i fem år och använder det knappt halva sin arbetstid. Första kalkylsystemet var utformningen av en personalbudget. Sedan har kalkylprogram framför allt använts för beräkningar av typerna *Ordbehandlarsystemet* och *Lilla kalkylsystemet*. Jan har ett femtiotal kalkylsystem som han använder regelbundet under året. I genomsnitt skapar han ett nytt kalkylsystem två gånger i veckan. En mindre del av dessa är tillfälliga beräkningar som inte sparas. Viktiga användningsområden är utformning av budget, årsredovisning och rapportering av olika slag. Enligt egen utsago arbetar Jan gammalmodigt och "...skulle vilja kunna fler finesser för att bli mer effektiv." Jan har gått en del internkurser i kalkylprogram samt en kurs i ekonomistyrning med orientering mot byggande av kalkylmodeller. Mellan kalkylsystemen som Jan gjorde när den första intervjun gjordes 1995 och kalkylsystemen som fanns då den senaste intervjun gjordes 1998 märktes en viss ökning av komplexitet.

Jan ser sin datoranvändning i ett helhetsperspektiv. Användning av kalkylprogram, ordbehandlingsprogram, databashanteringsprogram och centrala system är en del av hans utförandet av arbetsuppgifter.

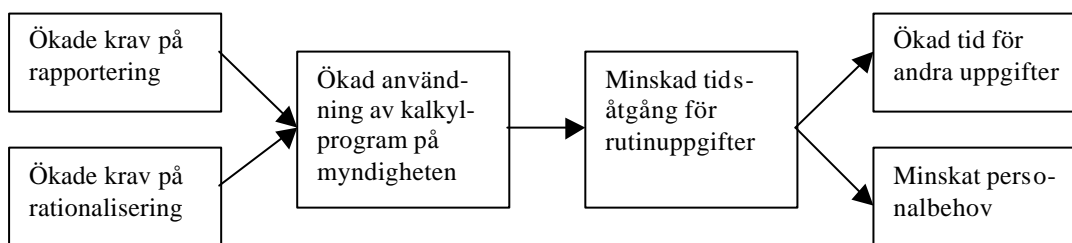
4.4.4 Anveckling, exempel och analys

Rationalisering

Jan har framför allt ett rationaliserande syfte med sin anveckling. Han ser kalkylprogrammet som ett hjälpmedel för att utföra arbetet snabbare och säkrare. Målet är att

myndigheten skall förbruka mindre resurser och använda mindre antal personer till rutinarbete. Ett exempel är budgetarbetet som "...går oerhört mycket snabbare nu, jag tror inte att jag hade klarat det utan kalkylprogram". Den överblivna tiden sugts för hans del upp av annan verksamhet. Jan strävar efter att denna tid skall kunna användas till ekonomistyrning, som i sin tur syftar till att stödja en mer ändamålsenlig verksamhet. Det har under hans tid på myndigheten inte varit en statisk situation. Organisationen har genomgått omfattande förändringar sedan början på 1990-talet. Effektiviteten i form av mer rationellt arbete har utnyttjats till verka inom utvidgade ansvarsområden. De nya ansvarsområdena har i sin tur medfört nya uppgifter som medfört behov av datorstöd, bl.a. med kalkylprogram.

Användning av kalkylprogram har fungerat, som ett sätt att rationalisera arbetet för att dels uppfylla målet med resurseffektiv verksamhet och dels kunna utföra nya uppgifter som förväntas utföras av myndigheten. En ökad rapportskyldighet har också ökat behovet av datorstöd (se Figur 55 nedan).



Figur 55 Förutsättningar för och effekter av kalkylprogramanveckling på myndigheten.

Ett första kalkylprogrambaserat **budgetsystem** byggdes 1993. Den bestod huvudsakligen av enkla summeringar och avstämningar. I den andra budgeten från 1994 introducerades länkning, vilket möjliggjorde en mer avancerad form av simulering än vad som var möjligt i den första versionen. En ändring i en delbudget, t.ex. personalbudgeten, slog omedelbart igenom på hela budgeten om länkarna var riktigt utformade. Tidigare var det "...ett helvetiskt räknande". Problem som kunde uppstå vid budgeteringsarbetet med kalkylsystemet var då budgetens struktur skulle förändras. Länkningsar och formler var tvungna att utformas så att summorna blev rätt om man t.ex. infogade en ny rad. Även s.k. namn i formler användes. Fördelen med s.k. namn i formler är att formlerna blir delvis självdokumenterande.⁵ Vid utarbetandet av kalkylsystemet fick Jan hjälp av en kollega med goda IT-kunskaper. Vid arbetet med 1996 års budget användes version 5.0 av MS Excel där det var möjligt att dela upp ett kalkylsystem på flera blad. Budgetsystemet fick då plats i en arbetsbok med ett blad per avdelning och ett sammanställningsblad. Mellan bladen finns länkar. Även här upplevde Jan en viss osäkerhet i att veta om länkarna var kompletta och pekade rätt.

De fördelar som Jan upplever med kalkylsystembudgetering är att det "...blir tydligt dokumenterat, jämfört med kollegieblocket. Sen så räknar det automatiskt. Med jämförelse med innan jag fick Excel 1993, så blev det förbättring. Ett otroligt automatiseringshjälpmedel. Jag lägger ner mycket

⁵ En formel med namn kan t.ex. formuleras =Intäkt-Kostnad i stället för t.ex. =F21-E21.

Myndighet

mindre tid på att räkna nu. Ja, det har varit en jätteautomatisering, en jätterationalisering."

Om systemet har använts en gång tidigare, känns det säkrare att resultatet blir rätt.

Beslutsunderlag

Ibland får Jan till uppgift att förse kommunstyrelse, byggnadsnämnd eller någon annan myndighet med information av något slag. Ett exempel på det är rapport om **investering per verksamhet** (se Figur 56 nedan). Rapporten innehåller en uppställning av olika uppgiftstyper per verksamhet. Summaraden och nettokolumnerna innehåller formler som sköter beräkningarna. Kalkylsystemet kan sägas ha följande egenskaper

- systemet kan återanvändas om förfrågan skulle återkomma
- systemet kan verifieras genom att man matar in enkla data
- systemet kan fungera som undersystem till ett annat kalkylsystem om så skulle vara aktuellt (med hjälp av länkning)
- systemet kan hämta data från andra undersystem om sådana finns (med hjälp av länkning)
- systemet kan omformateras om anvecklaren skulle föredra en annan layout
- systemet kan visas i olika vyer om det skulle vara så att olika användare efterfrågade olika typer av information
- systemet kan visas i olika vyer om det av sekretessskäl skulle vara olämpligt att visa all information
- systemet kan distribueras till andra handläggare (ev. med cellskydd på vissa celler)
- systemet kan fungera som underlag till andra liknande system
- systemet kan, om det lagras i ett nätverk, användas av andra handläggare

Egenskaperna ovan kan sägas gälla för alla anvecklade kalkylsystem.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2	Investering per verksamhet												
3													
4													
5													
6	V-het Investeringar exkl fastighetsförsäljningar och explverksamhet												
7													
8		Bruttoinv	Inkomster	Nettoinv		Inkomster	Fastighetsförsäljningar	Explink	Explutg	Explnetto		Totalbokf investutg	investink
9													
10	9	29,2						29 216,0	14 576,7	14 639,3			
11	2	28 802,0	3 849,9	24 952,1		6 298,3	20 448,4						
12	3												
13	4												
14	5												
15	6												
16	8	952,4	0,0	952,4		9 957,3	9 957,3						
17	Summa	29 783,6	3 849,9	25 904,5		16 255,6	30 405,7	29 216,0	14 576,7	14 639,3		44 360,3	45 471,6

Figur 56 Rapport om investering per verksamhet.

Ett annat exempel på kalkylsystem är ett system för **beräkning av kreditivräntor** (ränta under byggtiden). Jan menar att om detta system inte funnits skulle kostnadsberäkningen utförts med sämre kostnadsunderlag. Detta skulle i sin tur ha medfört sämre ekonomisk hushållning och ett större risktagande med kommuninnevärnarnas pengar.

Till bokslutet har Jan tillverkat flera kalkylsystem som tillsammans kan sägas utgöra ett **bokslutssystem**. Själva bokslutet beräknas med hjälp av kalkylsystem och flera bokslutsbilagor är också kalkylsystem. Presentationen av bokslutet har Jan också utfört med hjälp av kalkylsystem i form av tabeller och diagram härledda ur själva bokslutet. Möjligheterna till presentation kan underlättas av att uppgifterna finns i kalkylsystemet, vilket är ett sätt att uppfylla målet för öppenhet mot kommunstyrelse och kommuninnevånare.

Externa och interna data

Ett system som Jan använder är **beräkning av fastighetsskatt** på kommunens fastigheter (se Figur 57 nedan) fungerar på så vis att man först hämtar en stomme av fastighetsuppgifter från ett centralt fastighetsregister. Sedan tillfogas en kolumn med skattesats och en med formel för beräkning av skattebeloppet. Systemet är ett exempel på hur externa data kan utnyttjas för och integreras med lokala bearbetningar. Systemet innebär en säkerhet i beräkning av skatt och en möjlighet för Jan att när som helst kunna veta exakt vad skattekostnaden blir.

	A	B	C	D	E	F	G	H	I	J
1	TID	TYP-	Församlin	Fastighet	SUMMA	Mark-	Bygg-	Sa markv	Skatte-	Skatt
2		KOD			TAXV	värde	värde	o byggv	sats	
3	18011019900002	120	ADOLFSB	GODSLAG	8 934	148	220	368	1,70%	6 256
4	18011019900002	120	ADOLFSB	GODSLAG	8 934	148	220	368		
5	18011019900002	120	ADOLFSB	GODSLAG	8 934	148	220	368		
6	18011079910002	120	ALMBY	ALMBY 11	6 804	182	416	598	1,70%	10 166
7	18011079910002	120	ALMBY	ALMBY 11	6 804	182	416	598		
8	18011079910002	120	ALMBY	ALMBY 11	6 804	182	416	598		
9	18011079910002	120	ALMBY	ALMBY 11	6 804	182	416	598		

Figur 57 Del av kalkylsystem för beräkning av fastighetsskatt.

I andra tillämpningar, som är utvecklade på myndigheten är fastighetsregistret kopplat till en karta (GIS-system) där det är möjligt att se hur fastigheterna ser ut och var de ligger.

Standardisering

En förändring i Jans arbete under de senaste två åren, är förekomsten av kalkylmodeller, som kommer på disketter från kommunkontoret. I dessa system, som syftar till att smala in information från myndigheterna i olika frågor, finns färdiga blanketter, som man bara kan fylla i. Större delen av Jans arbetstid som berör kalkylsystem läggs ned på system som han bygger själv. Standardblanketterna gäller de uppgifter som andra kontor skall ha, t.ex. ekonomikontoret, de kommer enligt Jan att bli mer och mer standardiserade. Jan menar att det inom det kommunala är mycket förändringar, nya lagar och nya regler nya hänsyn till miljön, som gör att man måste ompröva modellerna och att man måste göra nya budgetmoduler, nya krav på redovisning och presentation. En del av dessa krav kan tillgodoses med standardmallar medan andra måste lösas med nya kalkylsystem. Merdelen av rapporterna skapas genom byggande av egna kalkylsystem. Ytterligare ett exempel på ett sådant eget kalkylsystem är **redovisning av kostnader** med eller utan inräknande av internttransaktioner (se Figur 58 nedan). Genom

Myndighet

detta kalkylsystem kan myndigheten visa ett mer rättvisande resultat eftersom interntransaktioner inte är att jämföra med transaktioner med externa intressenter.

	A	B	C	D	E	F	G	H	I	J	K	L
1	BRUTTOREDOVISNING MED OCH UTAN EGEN MOTPART											
3			Utän egen motpart				Med egen motpart			Avvikelser		
4			Intäkter		Kostnader	Netto	Intäkter	Kostnader	Netto	Intäkter	Kostnader	Netto
5	Företag 75											
7	Planering och myndighetsutövning		-13 348,1		35 134,3	21 786,2	-26 336,7	43 408,2	17 071,5	12 988,6	-8 273,9	4 714,7
10	Kollektivtrafik		-16 150,4		71 830,7	55 680,3	-16 150,4	72 094,0	55 943,6	0,0	-263,3	-263,3
12	Särskild kollektivtrafik		-36 691,9		36 459,1	-232,8	-36 697,8	36 659,3	-38,5	5,9	-200,2	-194,3
14	Park o rekreation		-2 243,2		18 176,1	15 932,9	-2 450,3	19 214,7	16 764,4	207,1	-1 038,6	-831,5
16	Bost pol åtgärder		-789,0		9 317,6	8 528,6	-832,2	9 869,3	9 037,1	43,2	-551,7	-508,5
18	Mark		-34 772,2		42 168,9	7 396,7	-37 238,0	44 744,3	7 506,3	2 465,8	-2 575,4	-109,6
20	Tot företag 75		-103 994,8		213 086,7	109 091,9	-119 705,4	225 989,8	106 284,4	15 710,6	-12 903,1	2 807,5
22	Överföring till investeringsredovisning											2 607,9
24	Differens											199,6

Figur 58 Del av kalkylsystem för rapportering av bruttoredovisning med eller utan interntransaktioner.

Andra verktyg

På avdelningen används även databashanteringsprogrammet Access i ökad utsträckning. Personalens tidsskrivningssystem på myndigheten är t.ex. konstruerat med MS Access. I systemet finns tidsuppgifter per person och projekt. Jan har önskemål om att automatisera åtkomst till detta system. En ökad användning av Access kan också märkas på myndigheten. Jan tycker också att det är bra att kunna länka mellan MS Word och MS Excel eftersom man då kan ha samma källdata.

4.4.5 Arbetssätt

Ett önskemål som Jan haft under flera år är *möjligheten att kunna hämta data från centrala system*. Speciellt finns önskemål om direktåtkomst till ekonomisystemet, för att hämta information till egna bearbetningar. Viss åtkomst till centrala data finns, t.ex. till centrala fastighetsregistret i Gävle. Delar av, det för myndigheten viktiga, centrala fastighetsregistret tankas regelbundet ner till myndigheten, vilket ger möjlighet till åtkomst av information. Ett problem för anvecklingsverksamheten är följaktligen begränsad tillgång till information i centrala system.

Ett annat problem som Jan noterat är *upplevd otillräcklighet när det gäller kunskaper om kalkylprogram*. Jan tror att han skulle kunna arbeta mer effektivt om han kände till mer finesser och genvägar om kalkylprogramanvändning.

En nackdel med *återanvändning av tidigare kalkylsystem* är att gamla siffror (och formler) kan bli kvar och ge felaktigt resultat. För att upptäcka den typen av fel måste informationen kontrolleras av personer med verksamhetskänedom.

Jans arbetssätt vid kalkylprogramanveckling är av *hands-on* karaktär. Inga specifikationer eller skisser görs före byggandet av kalkylsystemet. "Det är siffrorna som gäller." Om dokumentet skall distribueras kan Jan formatera det i efterhand, när beräk-

ningarna är utformade. En viktig aspekt är *verifiering* av systemets utinformation. Denna verifiering underlättas om systemet använts tidigare.

4.4.6 Sammanfattning - Jan

Jan är positiv till användning av kalkylprogram. Han använder det ofta som ett verktyg i sitt arbete som ekonom och ser tidsbesparing som det främsta syftet. Den överblivna tiden kan användas till personalbesparingar och utförande av andra icke rutinmässiga uppgifter.

Kalkylprogramanveckling bidrar till att uppfylla mål i verksamheten. I relation till *Etik för kommunalekonomer* (se *Bilaga 6*) kan kalkylsystem, som Jan har byggt, sägas

- Verka för god ekonomisk hushållning (regel 2 och 6)
- Verka för att relevanta fakta om verksamheten inte döljs (regel 6)
- Verka för saklig beredning inför beslut (regel 7)
- Bidra till medvetande om ekonomiska konsekvenser (regel 7)
- Verka för öppen och korrekt information (regel 9)
- Minimera risktagande (regel 11)

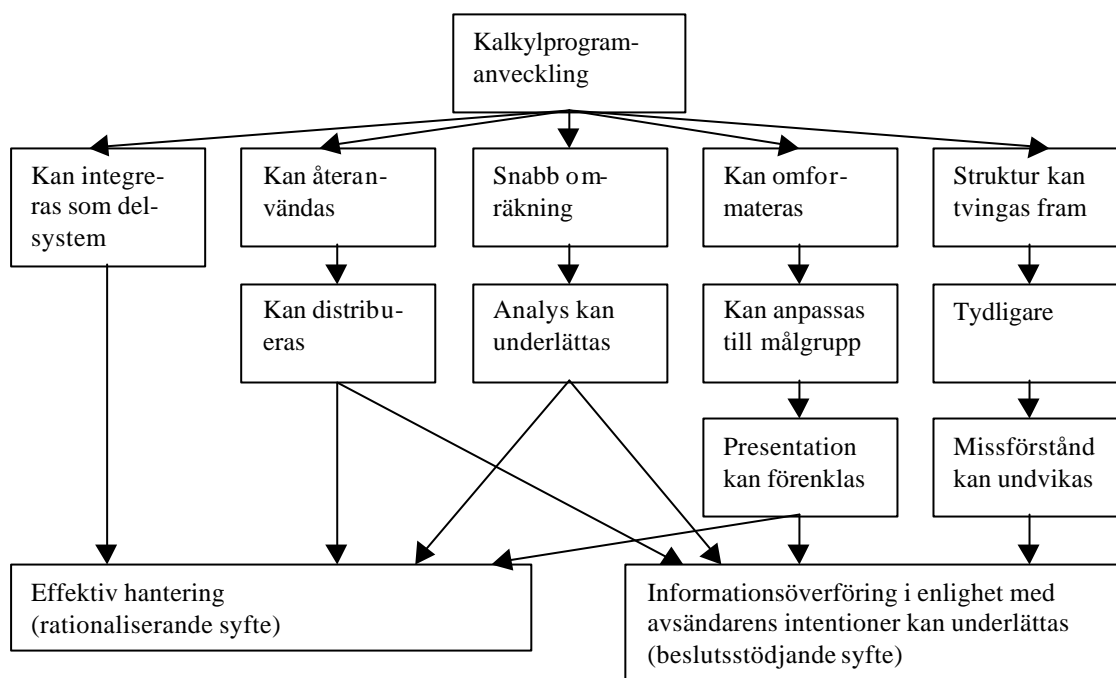
Målet för kommunala myndigheter att prestera ett nollresultat underlättas också av en rationell hantering av myndighetens ekonomi.

Följande effekter kan relateras till Jans användning av kalkylsystem (se Figur 59 nedan):

- Möjligheter till **presentation** underlättas
- **Tydlighet** kan innebära mindre missförstånd än manuella system
- Permanentningen av kalkylsystemet innebär att **återanvändning** och **integration** med andra system kan ske
- Analys underlättas av **snabb omräkning**

Dessa egenskaper kan bidra till såväl rationaliserande som beslutsstödjande syften.

Myndighet



Figur 59 Effekter av kalkylprogramanveckling.

4.5 Annette - administratör

4.5.1 Uppgifter och verksamhetskunskap

Annette har under tre år arbetat som miljö-, kvalitets- och informationsansvarig på myndigheten, med tonvikt på de senare två uppgifterna. Var tidigare bygglovsadministratör, dock ej med handläggning som huvuduppgift, utan "...pappersexercis" - hantering.

Annettes personliga förhållningssätt till datoranvändning utmärks av motivation, envishet och att hon tycker att det är roligt att arbeta med datorer och prova nya sätt att lösa uppgifter.

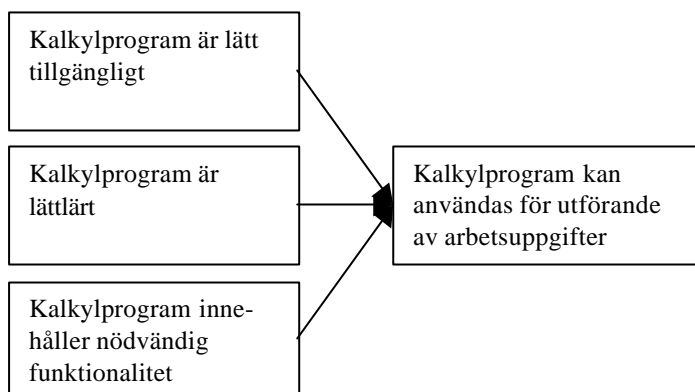
4.5.2 Mål

Målen för Annettes verksamhet är de som gäller för myndigheten, vilka beskrivs i inledningen av kapitlet.

4.5.3 Anveckling, frekvens och typ

Annette har använt kalkylprogram vid två tillfällen, dels för att lämna bygglovsstatistik till Statistiska Centralbyrån (SCB) och dels för att sammanställa en personalenkät på myndigheten. Viktiga allmänna egenskaper för ett utvecklingsverktyg i detta sammanhang var, förutom att det skulle innehålla nödvändig funktionalitet, att det fanns lätt tillgängligt och var lättlärt. Skälet till att detta var viktigt var att det inte var aktuellt

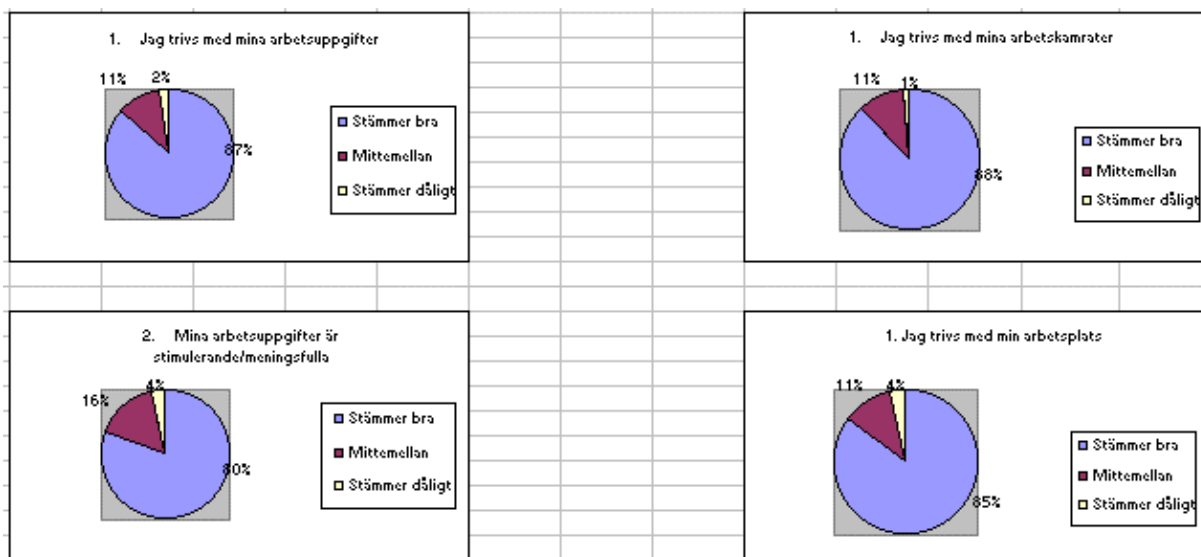
vare sig att införskaffa något annat verktyg än de som redan fanns på myndigheten eller att anlita extern kompetens för att utveckla system. Utvecklingsverktyget skulle kunna fungera som ett verktyg för verksamhetsföreträdare som behövde datorstöd för att utföra sina arbetsuppgifter (se Figur 60 nedan).



Figur 60 Motiv för användning av kalkylprogram.

4.5.4 Anveckling, exempel och analys

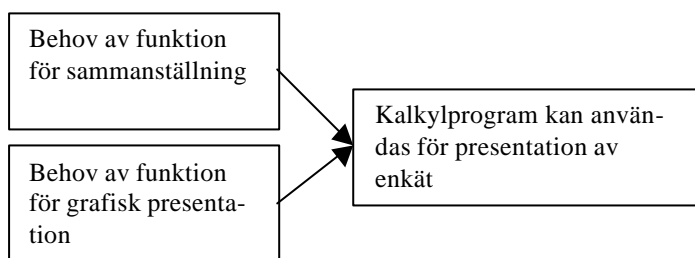
Det första systemet som Annette gjort med kalkylprogram är ett system för **sammansättning av en personalenkät**. Enkäten sammanställdes först på kalkylblad fråga för fråga. Utifrån sammanställningen skapades sedan ett cirkeldiagram för varje fråga (se Figur 61 nedan).



Figur 61 Del av system för presentation av resultat av personalenkät.

Systemet för personalenkät gjordes utan tidigare kunskaper i kalkylprogram. Annette gjorde trots detta systemet utan hjälp, vilket inte var utan problem. Speciellt upplevdes problem när det gällde att formatera diagram. Efter en hel del målmedvetet experimenterande blev resultatet ungefär som Annette avsett från början. Skäl till att kalkylprogramvaldes var bl.a. att det fanns behov av ett verktyg med funktioner för samma-nställning och presentation (se Figur 62 nedan).

Myndighet



Figur 62 Motiv för verktygsval för system för personalenkät.

Det andra systemet som Annette har gjort, hade till syfte att skapa **statistik över bygglovsärenden**. En bygglovsdatabas innehåller alla bygglovsärenden. Det kan gälla allt från stora byggen till skyltar på polishuset. Alla handlingar finns registrerade, inklusive avslag, ej beviljade, information om byggherre, fastighet, ärende, besiktning.

Syftet med systemet är att skapa en rapport, som skulle skickas till Statistiska Centralbyrån (SCB), eftersom Sveriges kommuner är skyldiga att förse SCB med uppgifter av denna typ. Kalkylsystemet utnyttjar ett frågeverktyg, som är integrerat i kalkylprogrammet,⁶ och ersatte ett tidigare system i minidatormiljö där man fick konstruera SQL-frågor mot en minidatorbaserad databas. Då databasen konverterades till PC-miljö uppstod möjligheter att ställa frågor mot databasen på ett enklare sätt. I stället för att skriva SQL-frågorna kunde man klicka, peka och dra för att formulera frågorna.

	A	B	C	D	E	F	G	H	I	J	K	
1	LopNr	Ar	Placering	Version	ansdat	besdat	dpb	dpb1	forsaml	fastbet	fastadr	sokandel
2	1826	96	AN	1	1995-11-01 00.00.00	1995-11-08 00.00.00	4444		02	OLAUS PETRI 3:33	S GREV ROSENGATAN 33	ABB
3	1826	96	AN	1	1995-11-01 00.00.00	1995-11-08 00.00.00	4444		02	OLAUS PETRI 3:33	S GREV ROSENGATAN 33	ABB
4	1830	96	AN	1	1995-11-01 00.00.00	1995-11-15 00.00.00	4443		02	OLAUS PETRI 3:33	S GREV ROSENGATAN 33	ABB

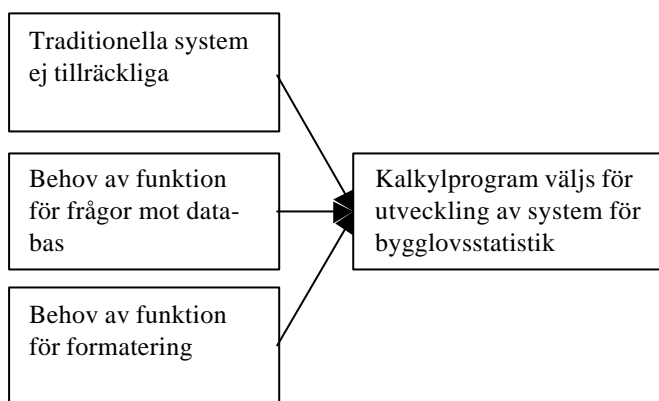
Figur 63 Del av system för analys av bygglov.

Systemet fungerar som så att man från kalkylprogrammet aktiverar frågeverktyget och formulerar frågor mot bygglovsdatabasen. När frågan formulerats, exekveras frågan och svaret returneras till ett kalkylblad (se Figur 63 ovan). Då svaret matats in på kalkylbladet utförs viss formatering, varefter bladet skrivs ut och skickas till SCB.

Ett problem som upplevts med systemet gäller prestanda. Systemet var lite väl långsamt. Vissa oklarheter i hanteringen av frågorna fanns också. Gamla frågor kunde ibland vara förstörda, vilket kunde bero på att frågan avbrutits då den exekverats den senaste gången. Ett annat problem var att formatering tidigare inte sparades i kalkylbladet.⁷ Skäl till att kalkylprogram valdes var bl.a. att det fanns behov av ett verktyg med funktioner för databasfrågor och formatering samt att traditionella system ej var tillräckliga (se Figur 64 nedan).

⁶ MS Query som är integrerat i MS Excel.

⁷ Detta problem är åtgärdat i och med version 97 av Excel, som var utvecklingsverktyget.



Figur 64 Motiv för verktygsval för system för bygglovsstatistik.

4.5.5 Arbetsätt

Annette gjorde sina system utan tidigare kunskaper om kalkylprogram. Hennes arbetsätt kan klassificeras som hands-on eller trial-and-error. Eftersom hon hela tiden stötte på problem, krävdes en hel del envishet för att lösa dessa. Ibland fick hon gå omvägar om det inte gick att göra som hon tänkt sig från början.

4.5.6 Sammanfattning - Annette

Annette har använt kalkylprogram som ett verktyg för att utföra specifika arbetsuppgifter i sitt arbete. De främsta motiven är att kalkylprogram varit **tillgängligt**, ansetts **lättlärt** och ansetts ha nödvändig **funktionalitet**. Med nödvändig funktionalitet avses möjlighet att ställa frågor, möjlighet till formatering, möjlighet till sammanställning och möjlighet till grafisk presentation.

Sammanfattning - Myndighet

Verksamheten på myndigheten utmärks av sin kommunala karaktär, vilket innebär att kommunala beslut och kommunala målsättningar styr verksamheten. De kommunala målsättningarna innebär att verksamheten varken skall gå med vinst eller förlust. Vidare innebär det att verksamheten skall bedrivas med kommuninnevånarnas bästa som mål. Verksamheten skall vara öppen för insyn och rättvis, på så vis att den skall följa lagar och regler. Förändringar i regelverk, omvärld och IT-miljö innebär att anveckling krävs för att upprätthålla målsättningarna under ändrade förutsättningar.

Kravet på öppenhet innebär att beslutsunderlag skall finnas tillgängligt och vara begripligt, vilket gör att kalkylsystem kan användas som dokumentation av beräkningar. Kravet på rationell hantering inverkar på ekonomihanteringen, där Jan har rationaliserat bort en del manuell hantering med hjälp av kalkylprogram. Även Olle använder kalkylprogram för rationalisering. Kalkylprogram används även som verktyg för beslutsstöd.

Myndighet

Kalkylprogram har använts på flera olika sätt. De tre anvecklarna har haft varierande behov och varierande nytta av kalkylprogram. Olle och Jan har använder kalkylprogram mycket frekvent medan Annette använt kalkylprogram vid enstaka tillfällen. Jan ser rationalisering som huvudsyftet med sin anveckling medan Olle har den mest varierade användningen av kalkylprogram. Han har byggt kalkylsystem av olika komplexitet och med olika varianter av formalisering. Olle fungerar även som internkonsult på myndigheten och har byggt flera kalkylsystem som används av kollegor till olika ändamål.

"Nästa månad behöver man flera rader. Man får trixa, skjuta in rader, det tar tid. Det blir inte flexibla lösningar. Det är en nackdel att användaren inte är programmerare, men det är en fördel att felet upptäcks. Användaren vet ju när det är fel och när det är rätt."

Hans Ingwald, IT-chef Sverige, Dyno Nobel (Avdic 1999:117)

- *Detta kapitel syftar till att redovisa tre IT-specialisters erfarenheter av kalkylprogramanveckling med speciellt fokus på problem i samband med kalkylprogramanveckling.*

5 IT-specialister

Undersökningen syftar speciellt till att ta fram nackdelar med kalkylprogramanveckling, som inte kommit att belysas i de första två undersökningarna (Industri och Myndighet). Skälet till att intervjua IT-specialister är att de anvecklare som intervjuats kan förmodas vara mer positiva än negativa i och med att de fortsatt att använda kalkylprogram. De anvecklare som slutat eller inte börjat använda kalkylprogram kan vara svåra att hitta. Ett sätt att fånga upp ifrågasättande synpunkter på kalkylprogramanvändning skulle därför kunna vara att intervjua personer med erfarenheter av kalkylprogram från ett IT-specialistperspektiv.

5.1 Genomförande

För allmänna principer för metodanvändning hänvisas till kapitel 2. I detta avsnitt redovisas endast delprojektspecifika metodaspekter.

5.1.1 Metod

Undersökningen som genomförts, kan, liksom undersökningen på industrin och myndigheten, betecknas som en kvalitativ fallstudie genomförd med (1) öppna intervjuer och (2) informella samtal som genomförts vid varierande tillfällen.

Intervjuerna (1) har följt intervjumallen i *Bilaga 3* och gjorts på samma sätt som redovisats för myndigheten i föregående kapitel. Registreringar av de informella samtalen (2) har även de gjorts på samma sätt som redovisats för myndigheten.

5.1.2 Urval av respondenter

Urvalet av respondenter har gjorts för att fånga in olika specialistperspektiv. **Hans** är IT-chef, **Sten** är projektledare och systemutvecklare och arbetar även med användarstöd. **Mika** är konsult med bl.a. kalkylprogram som sin specialitet.

Respondenterna har jag lärt känna på olika sätt. Hans har jag tidigare haft ett utvecklings-samarbete med. Sten och Mika är f.d. studenter som jag lärt känna som lärare.

5.1.3 Disposition av presentation

Presentationerna av respondenterna och deras erfarenheter följer följande disposition:

- beskrivning av respondenternas arbetsuppgifter och verksamhetskunskap
- för Hans och Sten har verksamhetens IT-miljö presenterats (vilket inte varit aktuellt för Mika som arbetar som konsult)
- presentation av redovisade problem med KPA
- sammanfattning

Ej refererade citat i texten är avskrifter från bandinspelningarna.

I presentationen av intervjuerna har termen användare använts i stället för termen utvecklare. Skälet till detta är att respondenterna använt det uttrycket och att det kan innebära förvanskning av utsagor, att byta ut ett av de i intervjuerna vanligast förekommande orden. Av skäl som redovisats i kapitel 1 anser jag dock fortfarande att begreppet användare är i viss utsträckning missvisande även om det är den i särklass vanligaste benämningen på den grupp av personer som är i fokus för denna avhandling.

5.2 Hans - IT-chef

Hans arbetar som IT-chef för region Sverige, vid Dyno Nobel, vars affärsidé är "att lossöra berg". Företaget sysslar med att tillverka sprängämnen och tändmedel, tillhandahålla konsulttjänster, bedriva anläggningsförsäljning och joint venture (försäljning av halvfabrikat).

Som IT-chef Sverige arbetar Hans med att driva IT-sidan Sverige, vilket innebär drift, utveckling och förvaltning. "Numera är det mycket standardlösningar, mindre utveckling. Inte så mycket skraddarsydda lösningar." På koncernnivå finns en avdelning, som sysslar med standardisering. Bl.a. har man beslutat att gå över till Lotus SmartSuite i sin PC-miljö från att tidigare ha haft MS Office. Hans fungerar också som IT-chef på det företag där han är stationerad och där han mestadels arbetar. Det är från detta företag som uppgifterna i detta avsnitt är hämtade.

Hans uppfattar att det övergripande målet med hans arbete är att stödja företagets affärsidé och företagsövergripande strategi. IT-avdelningens egna mål på företaget där Hans arbetar formuleras:

- Att prioritera områden för IT-verksamheten
- Att förbättra dialogen med användarna
- Fånga upp behov i verksamheten

För att uppfylla målen har IT-avdelningen gjort en enkät på om hur avdelningen uppfattas i avsikt att hitta något område att fokusera. Resultatet av undersökningen blev att kommunikation med internanvändarna i företaget skulle prioriteras.

Hans är själv kalkylprogramanvändare. Han har utvecklat några kalkylsystem för egen del, bl.a. ett budgetsystem. För en IT-chef är det inte så vanligt att bli användare av det system som han själv utvecklar. Hans använder inte kalkylprogram dagligen. Mest används kalkylprogram under budgetarbetet.

5.2.1 IT-miljö

På företaget finns en Windows-PC-miljö där kalkylprogram är standard för alla PC-användare (ca 200). Hans vet inte hur många av användarna som verkligen använder kalkylprogram. Många använder endast ordbehandling. Avdelningarna för ekonomi, produktion och marknad har de flesta kalkylprogramanvändarna.

På företaget används två kalkylprogramfabrikat, Lotus och MS Excel. Egentligen är det meningen att alla skall gå över till och enbart använda Lotus kalkylprogram. En del avancerade användare har dock haft svårt att konvertera såväl mentalt som tekniskt. T.ex. kommer budgetsystemet att bli kvar i MS Excel eftersom makron i systemet inte kan konverteras automatiskt. De måste skrivas om, vilket det inte finns tid till.

5.2.2 Åtkomst till centrala data

Problem med kalkylprogramanvändning kan uppstå när användare behöver data från centrala system, vilket är ett vanligt förekommande önskemål. Användare kan nå data via rapportgeneratorer i ekonomisystemet eller i produktionssystemet. I dessa fall hämtas data från ett dictionary, som ekonomisystemets databas och produktionssystemets databas är moduler i. På detta sätt kan man få upp rapporter på skärmen eller på papper. Vill användaren vidarebearbeta data måste han "klippa och klistra från skärmen". Ett annat sätt att få åtkomst till data är via ett fjärdegenerationsverktyg, t.ex. Powerhouse, som finns på företaget. Då måste dock användaren först göra en beställning till dataavdelningen, som genererar filer med önskade data. I ett pågående projekt planerar man att skapa ett Data Warehouse. Med hjälp av Data Warehouse räknar man med att användare ska kunna få direkt åtkomst till en databas med centrala företagsdata. Denna databas måste dock först läggas upp av dataavdelningen. Ytterligare en möjlighet till databearbetning för användare är Data Mining, som nyligen börjat diskuteras på företaget. Hans tror inte att de nya verktygen kommer att ersätta kalkylprogram, utan snarare överlappa.

Det finns enligt Hans flera skäl till att data måste läggas till rätta innan användare ges åtkomst till centrala data. Om det skulle vara fri åtkomst till de centrala databaserna

skulle det bl.a. innebära att användare måste ha vetskap om status i databasen för att förstå vilken information som data representerar. Om användare t.ex. håller på med lönekostnader så bör han veta att den uppdateras en gång per månad, och om han tar ut uppgifter strax före månadsslutet så bör han veta att de inte är helt aktuella. Användaren kan missbruka eller misstolka information om han inte känner status. Det kan vara svårt att veta vad allt betyder. Datamodellen som ligger till grund för den omfattande databasen inte alltid entydig. "Man drunknar i tabeller". Det finns, enligt Hans, ett behov av att aggregera information för att göra den tillgänglig för användare.

För att skapa en katalog- och nätverksstruktur som stödjer möjligheterna för användare att nå data. I nätverket finns t.ex. en företagsgemensam och en avdelningsgemensam nivå. Den övervägande delen av informationen är enkelriktad, dvs dataavdelningen ansvarar för lagring och uppdatering av data. Användare ansvarar dock för att viss information finns i nätverket. T.ex. uppdaterar ekonomiavdelningen kontoplanen på den företagsgemensamma nivån. Kontoplanen ligger för övrigt på kalkylblad, en miljö som man är van vid på ekonomiavdelningen. Ett annat system, som inte dataavdelningen ansvarar för, är ett mallsystem, som innehåller mallar för t.ex. brevskrivning. Systemet ligger i en databas, konstruerad med databashanteringsprogrammet Approach.

Följande aspekter om åtkomst till centrala data presenteras av Hans:

- Hur användare når centrala företagsdata är inte självklart.
- Data måste sammanställas
- Verktyg för access måste tillhandahållas
- Nya tekniker kommer fram
- Utformningen av nätverksstrukturen är viktig för möjligheter till dataåtkomst
- Användare kan ha ansvar för viss uppdatering av data

5.2.3 Användningsområden för kalkylprogram

Ett viktigt användningsområde för kalkylprogram är, enligt Hans, grafisk presentation.

"Det är det vi använder kalkylprogram till. Centrala system byts inte så ofta. De är inte up-to-date vad gäller presentation. Det är värdefullt att kunna göra presentationer med data från centrala system. Att läsa tabeller är inget bra sätt att presentera uppgifter. Handritade OH-bilder går inte, trovärdigheten minskar."

En nackdel med grafisk presentation menar Hans är att det ger möjligheter att lättare *manipulera* data. "Det gäller att kunna läsa skalorna så man vet vad diagrammen visar." En annan nackdel är att det kan bli för *tidsödande* att göra diagrammen. "Det kan bli för mycket kosmetika."

Användningsområden med kalkylprogram enligt Hans:

- Grafisk presentation är ett viktigt syfte med kalkylprogram

5.2.4 Problem med kalkylprogramanveckling

Hans ser såväl fördelar som nackdelar med kalkylprogramanveckling på företaget. På ekonomiavdelningen, där det finns frekventa kalkylprogramanvändare, upplever Hans att *användarna inte gör optimala lösningar*. Anledningen till detta är att slutanvändarna inte är skolade på samma sätt som programmerare och därför gör misstag. Ett exempel på sådana fel är att modellerna som byggs blir statiska och inte klarar förändringar.

"Nästa månad behöver man flera rader. Man får trixa, skjuta in rader, det tar tid. Det blir inte flexibla lösningar. Det är en nackdel att användaren inte är programmerare, men det är en fördel att felet upptäcks. Användaren vet ju när det är fel och när det är rätt."

En annan form av anveckling kan uppstå när användare engagerar *konsulter som bygger för avancerade kalkylsystem*, eventuellt med makron. Problem med dessa system är att de varit för komplicerade. Användarna vågar ibland inte tro på systemet då man inte kunnat utvärdera logiken i den komplicerade modellen.

Dålig eller obefintlig dokumentation av användares system är ett problem då personer byter jobb.

"Kompetensöverföring är svår. Dataavdelningen får ibland ta över system som slutanvändare gjort."

Användning av kalkylprogram utmärks i vissa fall av *begränsad prestanda*. Ett exempel på det är då stora datamängder skall bearbetas. Databasens begränsning i kalkylprogram har gjort att man inte kunnat använda kalkylprogram till vissa applikationer.

Nya versioner av anvecklarverktyg är ett problem för dataavdelningen, eftersom man strävar efter att alla skall ligga på samma nivå i hela organisationen. Om inte alla har samma version kan kompatibilitetsproblem uppstå. Vissa användargrupper (tekniker och personal på dataavdelningen) vill ofta ha senaste versionen. Eftersom versionsbyten är kostsamt, vad gäller programvara, maskinvara och utbildning, försöker man att "ta större kliv" och inte uppgradera vid varje versionsbyte.

Ytterligare ett problem är att *dataavdelningen inte hinner hålla sig uppdaterad på slutanvändarverktygen*.

"Vi försöker hänga med. Vi kan också slutanvändarverktygen lite grann men vi får inte så djup kunskap och det kan vara svårt att upprätthålla den förvärvade kunskapen."

Problem med kalkylprogram enligt Hans:

- Användarutvecklare utvecklar kalkylsystem som är svåra att ändra
- Kalkylsystem gjorda av konsulter kan bli för komplicerade
- Dålig dokumentation är problematisk vid byte av personal
- Kalkylprogram har begränsningar bl.a. vad gäller databasstorlek
- Nya versioner kan leda till ekonomiska, administrativa och kompetensmässiga problem

- Dataavdelningen hinner inte hålla sig uppdaterad på användarverktyg

Konsekvensen av problem med användare som utvecklar är enligt Hans att användarna själva blir lidande. Hans tror att lite mer avancerade makron skulle kunna eliminera en del av de problem, som finns med statiska kalkylsystem idag. Användarna skulle behöva ha viss programmeringskunskap.

Lösningar på problem enligt Hans:

- Mer programmeringskunskaper hos utvecklare skulle minska problemen med kalkylprogramanvändning

5.2.5 IT-avdelningen och verksamhetskunskap

På dataavdelningen arbetar man kontinuerligt med utveckling, förvaltning och drift av verksamhetsstödande informationssystem, som t.ex. ekonomisystem eller produktionssystem. Detta arbete ger god verksamhetsinsikt. Hans menar att det ibland har funnits mer verksamhetskunskap på dataavdelningen än på t.ex. avdelningen för produktionsplanering. Då ny personal har rekryterats, har inte dessa varit så insatta i företagets specifika verksamhet. De har varit experter på sitt yrke och för att få kunskaper om företagets verksamhet har man ibland fått fråga dataavdelningen.

"Eftersom dataavdelningen har jobbat så intimt med verksamheten så att vi får mycket kunskapsöverföring till oss. Om man kan systemet vet man vad som krävs i verksamheten."

- IT-avdelningen kan ha kunskap om affärsverksamheten som inte specialister på olika verksamhetsrelaterade områden, (t.ex. produktionsteknik) har.

5.3 Sten - systemutvecklare

Sten arbetar som projektledare och systemutvecklare vid AssiDomän Frövi, ett företag som sysslar med att tillverka pappersmassa och kartong. Som systemutvecklare sysslar Sten med nyutveckling och förvaltning. Dessutom arbetar Sten med användarstöd för PC-användare. Sten, som tidigare arbetat mest i stor- och minidatormiljö, har under senare år blivit allt mer positiv till PC-användning på företaget.

Sten har själv kommit att använda kalkylprogram i sitt arbete som projektledare. Han har gjort ett system med tidsplaner innehållande projekthändelser. Han har också utvecklat ett system för projektadministration med en arbetsbok per projekt (eller delprojekt) och där specifikationer på delsystem finns på blad i arbetsböckerna. Avvikelsehanteringen sköts också i kalkylsystemet. Presentation på beslutsmöten förenklas, enligt Sten, genom denna hantering.

5.3.1 IT-miljö

Företaget har en blandad miljö, med minidatorer och PC-nätverk med Windows NT både i servrar och hos klienter. Produktionen har ett eget datorsystem. Mycket databe-

handling ligger i standardsystem, som finns för bl.a. ekonomi, marknad, personal, underhåll och produktion. Alla användare som vill kan använda MS Office-paketet (version 97) med tillgång till MS Word, Excel och MS Power Point. Flera hundra användare är uppkopplade på det lokala nätverket.

5.3.2 Åtkomst till centrala data

Åtkomst till centrala data kan ske på flera sätt. På företaget har man byggt upp ett intranät som ger användare tillgång till olika delar av en intranätdatabas, beroende på vilken behörighet man har. Denna intranätdatabas består av rapporter av olika slag skapade i olika miljöer och lagrade i form av dokument. Dokumenten är skapade av olika personer i olika programmiljöer t.ex. databashanteringsprogram, kalkylprogram, och ordbehandlare. Flera av dokumenten är skapade i kalkylsystem, utvecklade av kalkylprogramanvändare. Detta innebär att verksamhetsföreträdare ansvarar för uppdatering av information som de har verksamhetsansvar för.

Det finns även möjlighet att nå traditionella databaser med SQL-frågor från applikationer. Denna användning kräver dock mer kvalificerade kunskaper.

5.3.3 Användningsområden

På företaget är MS Excel-användning vanlig. Användare kan, enligt Sten, delas in efter förkunskaper. Förkunskaperna varierar från användare som ser datorn som en "svart låda" till användare som skriver egna program. Användningsområden är simulering, (grafisk) presentation och sammanställning av data.

Sten upplever att *simulering* i form av s.k. what-if-funktioner är styrkan med kalkylprogram. Bl.a. marknadsavdelningen på företaget har utnyttjad denna funktion. Enstaka användare bygger system med hjälp av programmering. De flesta användare anlitar dock extern hjälp då programmeringskompetens behövs till kalkylsystem. Dataavdelningen har inte tid att ägna sig åt detta i någon större utsträckning.

Ett användningsområde för kalkylprogramanvändning är *presentation*. Presentationen kan vara textuell eller grafisk. Textuell information kan visa summeringar eller andra sammanställningar, medan grafisk information kan visas i form av diagram. "Det är lättare att ta åt sig bilder än siffror." "Det är ett sätt att sälja information." Ofta används kalkylprogram för att skapa presentationer som skall användas till OH-bilder. Sten tycker sig ha ändrat uppfattning och ser presentationsaspekter som viktigare nu än då han enbart arbetade i stor- och minidatormiljö.

Kalkylprogram används på företaget inte sällan för att *sammanställa och bearbeta data som hämtats från andra databaser eller system*. Företagets intranät är den främsta källan för dataåtkomst. Ett annat exempel på användning av centrala data, är när Sten vid ett tillfälle "lyfte ner" företagets hela faktureringsstatistik från företagets minidator till kalkylprogram för vidarebearbetning. Ytterligare ett exempel är användning av Excel för att bearbeta data från företagets databaser. En databas på företaget är särskilt

konstruerad för rapportering. Denna databas kan användare dels nå som en rapportdatabas, dels kan SQL-frågor ställas mot databasen i applikationer, t.ex. i kalkylsystem.

Kalkylprogram "...används för att det är *allmängods*". Det är "...lätt att gå igång". Sten har själv använt kalkylprogram för *prototyping*. Även om det ibland blir dåliga lösningar för att man börjar bygga systemet innan man tänkt igenom problemet, så menar Sten att "...om verktyget är tillräckligt snabbt, tjänar man på det ändå."

Användningsområden med kalkylprogram enligt Sten:

- Simulering
- Presentation
- Sammanställning av data
- Prototyping

Ett sätt att göra kalkylsystemen mindre känsliga är att utforma dem med parameterstyrning. Ett annat sätt är att utnyttja de funktioner som finns i programmet. Sten uppskattar speciellt följande funktioner i de senare versionerna av Excel:

- Filter (en funktion för urval ur tabeller)
- Delsumma (en funktion för att sammanställa och summera¹ tabeller)
- Pivottabeller (en funktion för att skapa interaktiva rapporter/korstabeller)

5.3.4 Problem med kalkylprogramanveckling

Problemen med kalkylprogram kan delas upp i nackdelar med själva kalkylprogrammet och nackdelar med användningen av det. Sten tycker att det är en dålig *datahantering* i kalkylprogram. Som exempel på det nämner han:

- Databasen i MS Excel är ytterst enkel jämfört med databashanteringsprogram. Det finns t.ex. en begränsning i antal rader
- Rollback fungerar dåligt i kalkylprogram. Om det blir ett strömavbrott blir det "mischmasch i data, speciellt i batchsnurror"
- Postlåsningsen är otillräcklig, om ett blad öppnas, låses filen.² På företaget har man haft problem med fleranvändning i ett kalkylsystem för reklamationer, som delats av flera användare

Användares användning av kalkylprogram betecknar Sten som "anarki". Det "...byggs okontrollerat. Det finns kundregister på flera ställen". En del har "...viktiga saker på hårddisk som kan krascha". Om inte nätverket struktureras med eftertanke kan problem med säkerhet och behörighet uppstå. Det finns en risk att "...andra kan förändra data". I sämsta fall så kan användare förlora data.

Dataavdelningen har inte engagerat sig mycket i Excelproblem. Det blir problem när anställda som har kalkylsystem slutar. Gamla system kan vara problematiska att förvalta, speciellt om de är komplexa. "Ju duktigare konstruktörer, ju svårare." Det finns

¹ Eller utföra annan aritmetik, t.ex. medelvärde, standardavvikelse eller frekvens.

² Från och med MS Excel95 finns möjlighet till fildelning och viss rollbackfunktion.

sällan någon *dokumentation*. ”Det är för lätt att bygga system. Man luras in i det här.” På företaget finns inga *riktlinjer* för anveckling. Sten känner inte till några exempel på hur sådana skulle kunna se ut.

Problem med kalkylprogram enligt Sten:

- Dålig datahantering
- Anarkistisk användning
- Dokumentation saknas
- Riktlinjer saknas

5.3.5 Framtiden?

I framtiden tror Sten att kalkylprogram kommer att finnas kvar på företaget. Det sker en allmän utveckling inom IT-området, som är dokumentorienterad, vilket bl.a. innebär att kalkyldokument kan var en typ av dokument som hanteras.

Sten tror att det kan ske en utveckling mot integrerade miljöer som tillhandahåller olika typer av funktioner snarare än olika typer av program. Istället för användning kalkylprogram kommer användare att använda kalkylfunktioner. Nya tekniker kommer att i framtiden kunna användas tillsammans med kalkylprogram. Som ett exempel på en sådan teknik nämner Sten Data Warehouse, en teknik som redan övervägts på en avdelning på företaget.

5.4 Mika - konsult

Presentationen av Mikas syn på användning av kalkylprogram skiljer sig lite från Hans och Stens, beroende på att deras verksamhet skiljer sig. Hans och Sten såg kalkylprogramanvändning utifrån sin egen verksamhet i sina respektive företag. Mikas arbete som konsult blir inte riktigt jämförbar med denna situation. Det är inte användning av kalkylprogram i konsultverksamheten som studeras utan Mikas erfarenheter som konsult i andra verksamheter.

Mika arbetar som systemutvecklare och utbildare på ett konsultföretag (Martinsson), som sysslar med hårdvaruförsäljning, utbildning och systemutveckling. Systemutvecklingsarbetet, som Mika bedriver sker i Microsoftprodukterna Excel, Access och Visual Basic.

Ett vanligt utvecklingsarbete för Mika är tillsammans med en kund/användare, som antingen inte kommit någonstans med ett utvecklingsarbete eller som påbörjat ett system och inte kunnat klara av att avsluta arbetet. ”Det är problem om användaren gjort något tidigare. Om kunden inte kommit någon vart tar vi hand om det.”

5.4.1 Problem med kalkylprogramanveckling

När kunden påbörjat arbetet med ett kalkylsystem kan det vara problem med:

- Kopplingar av länkar

- Makron
- Formler med beräkningar (kan bli för komplicerade, kan ej hålla reda på parenteser)
- Dokumentation saknas
- "Ser ut som hej kom och hjälp mig"

Ett exempel på ett konsultuppdrag gällde en situation där data skulle hämtas till ett kalkylblad från en minidator. Problem uppstod p.g.a. att alla data låg i ett blad. I bladet fanns "...länkar kors och tvärs". Användaren hade bl.a. skapat länkar till fastlåsta celler. För att reda ut problemen delade Mika upp systemet på flera blad och skapade dynamiska länkar med hjälp av funktionen LETAUPP³.

Länkning som är vanlig i kalkylsystem, kan låsa fast modellen på ett oönskat sätt. Funktionen LETAUPP kan i vissa fall vara till hjälp för att undvika fastlåsnings, men den kan bara användas på en begränsad mängd data. Ett annat problem, som gäller kalkylprogrammet MS Excel, gör att problem uppstår vid borttag av referensblad. Länken finns kvar internt i bladet och kan ibland inte tas bort, vilket leder till oönskade frågor då dokumentet öppnas.

Mika tycker inte att det är fel att användare använder makron. *Makron* kan spelas in eller skrivas, som vid programmering.⁴ Makron kan dock skapa problem när användaren skapar makron genom inspelning.⁵ Skälet till problemen är att det kan fungera bra vid inspelningen men det blir ej generellt. Man vill att det skall fungera i alla lägen, vilket det inte gör när man ändrar förutsättningar. Enligt Mika minskar behovet av makron med varje ny version av kalkylprogrammet.

Användning av *formler* kan också skapa problem. Formulerna kan bli stora och komplexa genom att "...användaren krånglar till dem." "En del användare använder SUMMA-funktion till allt, även till subtraktion." Ytterligare ett problem med formler är användning av konstanter i formler, vilket leder till att systemen kan vara svåra att förändra.

Avsaknad av *dokumentation* är ett problem då användare utvecklar kalkylsystem. Mika menar att det inte behövs så omfattande dokumentation. "Det räcker med att skriva hur det fungerar så att man kan använda det om man kommer tillbaka om ett år." Mika menar att mycket är självdokumenterande. Men att speciella saker bör dokumenteras. Till detta kan makrobladet eller själva kalkylbladet användas.

Problem som Mika uppfattar med kalkylprogram:

- Ostrukturerade system är svåra att använda och förvalta
- Länkning kan göra system ostrukturerade och oflexibla
- Inspelade makron kan göra kalkylsystem svåra att ändra

³ Se sid 323 för exempel på användning av funktionen LETAUPP.

⁴ Även en kombination av inspelning och kodskrivning är möjlig och vanlig då utvecklaren nått en viss färdighet.

⁵ Problem som skapas av att användaren skriver kod har inte behandlats här.

- Formler kan bli komplexa
- Avsaknad av dokumentation försvårar användning och förvaltning

Mikas strategi för att undvika problem vid utveckling av kalkylsystem:

- Använd först papper och penna
- Prototyping kan användas, men bara av användare med viss erfarenhet
- Dela upp systemet på blad
- Skapa länkar med funktionen LETAUPP
- Dela upp beräkningar i flera steg (lättare att se beräkningarna, dela upp i ett par tre kolumner, då blir det även lättare att felsöka)
- Dölj eventuellt beräkningskolumner

5.4.2 Problemorsaker och lösningar

Användare krånglar till det därför att de har för lite *kunskaper och erfarenheter*, både av strukturering och av verktyg. Användare kan dels ha svårt att dela upp data och dels kan de ha svårt att använda verktyget på ett ändamålsenligt sätt, t.ex. vad gäller att hämta data och länka data.

"De användare som har lite erfarenhet fokuserar på verktyget och fastnar i hanteringen. De kan få totalstopp i huvudet genom att de stirrar sig blinda på verktyget. På en kurs i kalkylprogram kunde en ekonomichef plötsligt inte räkna procent."

Det viktigaste för användaren, som vill lära sig att använda kalkylprogram för att utföra arbetsuppgifter, är att få erfarenhet av verktyget/kalkylprogrammet. Erfarenhet fås genom användning. Användaren lär sig kalkylprogram genom att bygga kalkylsystem, anser Mika. Inlärningen av kunskap om programvaror och datoranvändning kommer att bli enklare genom att kunskapsnivån ökar, eftersom datoranvändning blir vanligare och programvaror blir mer lättillgängliga. Mika tror att grundkurser av den typ som finns idag så småningom kommer att bli överflödiga. I de fall som verktygsutbildning bedrivs, bör den (än så länge) först ta upp grundläggande funktioner i kalkylprogrammet och sedan huvudsakligen knyta an till arbetsuppgifter i användarens verksamhet.

"Användarna skall öva på det som de sysslar med! Kursmaterial innehåller ofta övningar som knyter an till ekonomifunktioner. När man träffar tekniker kan de inte knyta an till det. Det är viktigt att knyta an till arbetsuppgifter. Det kan bli för svårt annars."

Att knyta an till verksamheten kan dock också vara svårt, eftersom det är svårt att hitta lagom komplicerade uppgifter. Ofta blir det ett för svårt problem, när man anknyter till den egna verksamheten. Som exempel anger Mika kunskapsbehov om kalkylprogram, för fyra kategorier av användare/arbetsuppgifter (se Tabell 6 nedan). Undervisning bedrivs enligt Mika bäst i små grupper med möjlighet till personlig undervisning.

Tabell 6 Kunskapsbehov för olika typer av arbetsuppgifter

Arbetsuppgifter	Kunskapsbehov kalkylprogram
Sekreterare	Blankettutformning
Ekonomer	Ekonomianalysfunktioner
Tekniker	Tekniska funktioner
Ledningsgruppen	Analysfunktioner

Ledningsgruppen ser Mika som en grupp som skulle behöva större kunskaper om kalkylprogram. Han tror inte att chefer har insett möjligheterna med kalkylprogram. Chefer kan behöva kalkylprogram för att bli oberoende. De kan bl.a. hämta data från databaser och analysera dessa data. Kalkylprogram kan fungera som ett slags *beslutsstödsverktyg* för ledningen på ett företag.

Mika använder alltid *databashanteringsprogram* (MS Access) för att lagra och hämta data. Han menar att det är lättare för att utforma rapporter och skärmbilder med Access. Access har av Mika använts som databasmotor och för att hämta och vända och vrida på data. Kalkylprogram kan enligt Mika vara enklare att använda vid offertsystem eller kalkylmodeller. Kalkylprogram är lämpligt när systemet skall innehålla beräkningar eller då syftet är analys av siffror eller grafisk presentation (diagram). Själv använder Mika både Access och Excel för kostnadsställedovisning i sitt arbete. Redovisningsapplikationen hämtar data från Access till Excel via ODBC för att där skapa grafiska presentationer över läget.

Vikten av *presentation* betonas, enligt Mika, mer av användare än av utvecklare. Utvecklare kan resonera: "Courier 10 räcker för rapporter". Genom kontakter med användare har Mika "...ändrat åsikt lite grann". Mika menar att diagram inte tillför så mycket utan att det kan vara psykologiska skäl till varför man använder diagram. Diagram kan väljas som presentationsform om presentatören vill manipulera. "Man kan vilja dölja eller framhäva. Syftet med val av presentationsform beror på presentationens syfte." Mika menar att det skulle underlätta att ha standards för grafisk presentation för att få jämförbar information. "Vid likartade presentationer borde det vara samma diagramtyp och samma mätmetod."

De frekventa *versionsbytena* som äger rum för kalkylprogram menar Mika minskar behovet av makroprogrammering. Nya versioner innehåller funktioner, som tidigare krävde programmering. En förändring i MS Excel, som gett goda effekter är införandet av arbetsböcker med blad som kan nås via flikar.

"Flikarna var ett lyft! De underlättar logisk uppdelning av data och möjligheten att skapa systempaket. Funktionaliteten ökar. Det blir möjligt att lägga in diagram, makron och dokumentation i samma dokument."

5.5 Sammanfattning - IT-specialister

Alla tre IT-specialisterna har rötter i traditionell systemutveckling. De använder kalkylprogram för egen del och är på så sätt också anvecklare.

Huvudproblemen med kalkylprogramanvändning:

- Bristande dokumentation
- Ostrukturerade applikationer
- Begränsad datahanteringen i kalkylprogram

Lösningar på problem:

- Utbildning på verktyg
- Versionsbyten medför ökad funktionalitet
- Strukturera kalkylsystemen

Användningsområden:

- Presentation
- Sammanställning (av centrala data)
- Beräkning/simulering

Övrigt:

- Versionsbyten kan innebära både för- och nackdelar
- Verksamhetskunskap kan finnas hos dataavdelningen

"Att budgetera manuellt är knöligt. Tänk Dig att man skall fördela 90.000 timmar på 80 kostnadsställen med papper och penna."
Tommy Petterson, tillf. underhållschef, AssiDomän Frövi (Avdic 1999:131)

- *Detta kapitel syftar till att redovisa förutsättningar för och effekter av anveckling av ett kalkylsystem av typen applikation. Detta sker i form av en undersökning av utveckling av ett distribuerat system för budgetering.*

6 Applikation

Det i denna undersökning studerade systemet, **distribuerad budget**, är av typen *Applikation*, vilket innebär att det är verksamhetsföreträdare som initierat och genomfört systemutvecklingsarbetet. Budgetsystemet som utvecklats, påminner om ett traditionellt system utvecklat med traditionella utvecklingsverktyg på så vis att det innehåller kod och att det har ett antal (ovana) användare. Skillnaden ligger i att systemet är initierat och till stor del genomfört av personer på ekonomiavdelningen på det studerade företaget. Det utvecklade systemet kan hänföras till typen applikation, vilket är en typ av anveckling som gränsar till mer traditionell systemutveckling. Exemplet skall ses som ett specialfall av anveckling.

6.1 Genomförande

För allmänna principer för metodanvändning refereras till kapitel 2, i detta avsnitt redovisas endast delprojektspecifika metodaspekter.

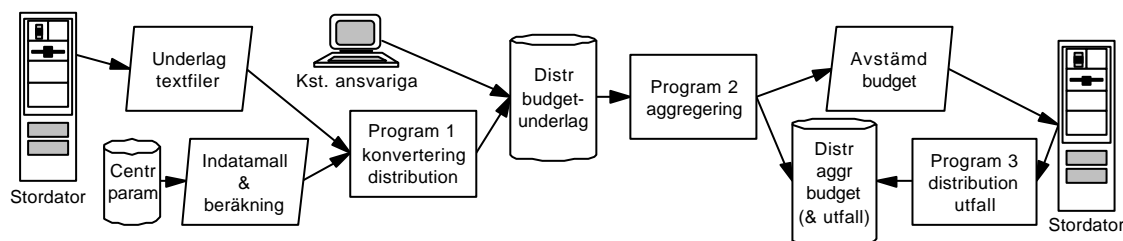
6.1.1 Metod

Undersökningen som genomförts, kan betecknas som en kvalitativ fallstudie huvudsakligen genomförd med deltagande observation samt öppna intervjuer dels med den budgetansvarige och dels med kostnadsställeansvariga. Intervjuerna har följt intervju-mallen i *Bilaga 4*. Intervjuerna är inspelade på band och därefter nedskrivna med hjälp av ordbehandlare. Den deltagande observationen består av att jag själv har deltagit i utvecklandet av systemet. Min uppgift har varit att utforma de program som behövs för att distribuera budgetar i företagets nätverk samt att utforma program för att sam-

manställa färdiga budgetar. Under arbetets gång har jag fått ta del av problem och andra omständigheter som jag subjektivt bedömt vara relaterade till problematiken med anveckling. Exempel på sådana omständigheter är då beställarnas verksamhetskunskap och yrkeskunskap styr utvecklingsarbetet eller då de kostnadsställeansvariga reagerat över systemet. Projektet har gett ett flertal tillfällen till reflexioner, som jag fått ta del av, hos de deltagande personerna, såväl beställare/systemägare som slutanvändare. Denna situation har, som jag upplever det, gett åtkomst till kunskap om omständigheter som vore svåra att observera med hjälp av en mindre ”närhetsorienterad” datainsamlingsmetod (se kap 2). Utöver observationer under projektets lopp har intervjuer med tre personer utförts efter att systemet användes första gången. De kostnadsställeansvariga har varit avdelningschefer. Dessa intervjuer har haft ambitionen att vara öppna, för att i möjligaste mån undvika förkategorisering. Intervjuerna har utgått från vilka för- och nackdelar som respondenterna upplevt med budgetsyste- met. Registreringen av intervjuerna har skett med bandspelare. Observationer gjorda under projektets gång har tecknats ned vid tillfällen som erbjudits. Då inte nedtecknande varit möjligt vid observationstillfället, har detta gjorts på kvällen. Kriteriet för att en observation skulle nedtecknats har varit att det, utifrån observatörens (jag själv) förförståelse, skulle kunna bidra till utformande av en anvecklargenerisk modell av den typ som beskrivs i kapitel 1. Utöver nämnda noteringar har systemdokumentation och systemet självt använts vid analysarbetet. Systemdokumentation har varit in- och utdatalayouter, programlistor, kravspecifikationer, testdokumentation och mötesanteckningar. Behandling av materialet har gått till så att intervjuanteckningar och systemdokumentation har registrerats med ordbehandlare. Under registreringen har noteringar av idéer och uppslag gjorts. Målet har varit att uppmärksamma mönster och kategorier som skulle kunna tänkas vara av betydelse för formulering av den anvecklargeneriska modellen. Bearbetning har skett genom upprepad läsning och omstrukturering, vilket har lett till att kategorier vuxit fram.

6.1.2 Miljö

Eftersom kalkylsystemet som utvecklats är av typen *Applikationen*, ingår programkod i systemet. Utvecklingsprojektet genomfördes mellan juli och november 1995, medan intervjuer och samtal har genomförts fram till slutet av 1997.

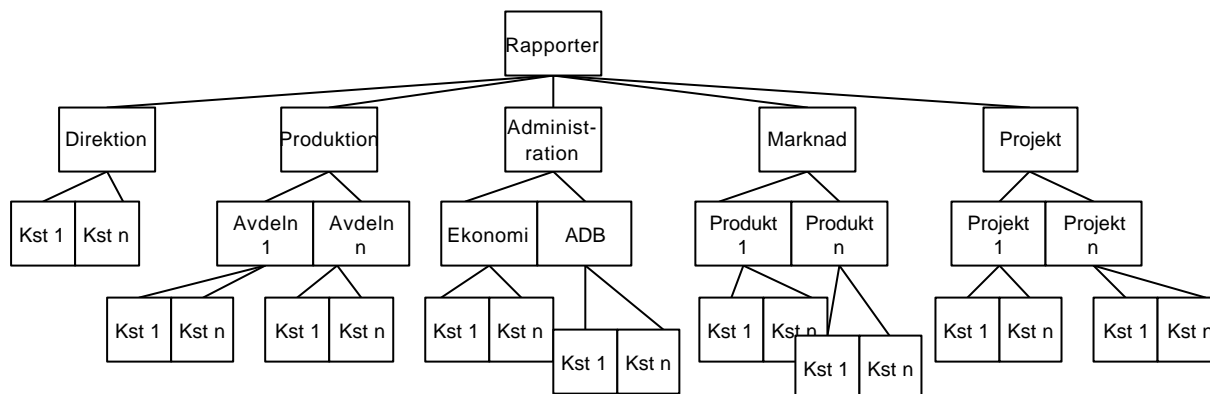


Figur 65 Schematisk beskrivning av budgeteringssystemet.

Figur 65 ovan visar systemets funktion schematiskt. Från stordatorsystemet genereras ett budgeteringsunderlag innehållande information om budget, utfall och differens från föregående år samt ackumulerat budget och utfall hittills under året. Underlaget kom-

Applikation

mer i form av en textfil, för att integreras i en budgeteringsmall, som är utarbetad av den budgetansvarige på företaget. Budgeteringsmallen innehåller tre blad med olika typer av information. I Figur 67 nedan visas ett exempel på ett inmatningsblad. På bladen finns kopplingar till en central parametertabell med värden, som fastställs av den budgetansvarige, t.ex. olika lönevärden och olika typer av procenttal.¹ Bladen innehåller även beräkningar och kopplingar mellan varandra. Program 1 konverterar² och integrerar textfilen samt distribuerar den i företagets nätverk till olika kataloger där den kostnadsställansvarige har läs- och skrivrättigheter. Figur 66 nedan visar schematiskt nätverkshierarkins uppbyggnad. Totalt finns ca 200 kostnadsställen.



Figur 66 Kostnadsställehierarkien schematiskt.

Då distributionen av budgetunderlag är klar vidtar själva budgeteringen som utförs av de ca 20 kostnadsställansvariga (slutanvändarna). Vid det första tillfället genomförde inledningsvis den budgetansvarige en utbildning där hanteringen demonstrerades och förklarades. Varje slutanvändare erhöll en till två timmars utbildning i grupper om ca 10 personer. Vid utbildningen fanns en dator tillgänglig för slutanvändarna att öva på. Under själva budgeteringen kunde den budgetansvarige då och då (ca 1 gång/vecka) sammanställa aggregerade budgetar på de olika nivåerna i hierarkin med hjälp av program 2. Detta gjordes dels av avstämningskäl och dels för att verifiera systemets funktion. När budgeteringen var klar sammanställdes alla budgetar och blev underlag för diskussioner och förändringar. När slutligen budgeten var godkänd av företagsledningen genererades en fil för inmatning i det stordatorbaserade ekonomisystemet. Under själva budgetåret (1996) har utfallssiffror (ackumulerat och för perioden) distribuerats tillsammans med jämförbara budgetsiffror. Distributionen har skett på liknande sätt som för budgetunderlaget.

I projektet kan urskiljas tre intressentgrupper:

1. Ekonomiavdelningen (inklusive den budgetansvarige). Det är denna grupp som initierat projektet och som ansvarar för att utdata är korrekt. Gruppen är *beställare* och *systemägare*. Gruppen kan även betraktas som anvecklare.

¹ Den centrala parametertabellen gör det också möjligt att i ett senare skede simulera aggregerade budgetar utifrån olika antaganden om verkligheten, t.ex. löner, inflation etc.

² Från textformat till kalkylbladsformat.

2. Kostnadsställeansvariga. Dessa personer är kalkylsystemets *slutanvändare*. Den genomsnittlige slutanvändaren var en manlig chef mellan 40 och 60 år och med liten datorvana.
3. Servicepersonal. Det är dels personal från dataavdelningen som sköter datorer och nätverk. I detta speciella fall har de skapat kataloghierarkin för kostnadsställen och tilldelat behörigheter till dessa kataloger. Dessutom är det en programmerare som gjort de tre programmen och deltagit i tester och provkörningar.

6.2 Resultat

Eftersom undersökningen är av kvalitativ karaktär och innehåller en stor mängd tolkningar är inte gränsen mellan resultat och diskussion väldefinierad. Nedan görs trots allt ett försök att göra en sådan gränsdragning.

Resultaten av undersökningen kan indelas i verksamhetsaspekter och systemaspekter. Verksamhetsaspekterna handlar t.ex. om huruvida mängden budgetinformation var lagom, om huruvida urvalet av information var det rätta. Dessa faktorer var synnerligen väsentliga för den budgetansvarige och utvärderades inför nästa års budgetering. Värderingen visade att informationen varit alltför detaljerad och planeringen inför budgeteringen innebar att information skulle presenteras på ett mindre detaljerat sätt. Dessa verksamhetsorienterade aspekter behandlas dock inte vidare här då de bedöms vara oberoende av huruvida budgetering och utfallsdistribution finns på papper eller kalkylblad.

Systemaspekterna handlar om faktorer som har med själva systemet att göra. Nedan görs en uppräkningslista utan prioritetsordning. Kategorierna har genererats induktivt under bearbetning av observationsdata.

6.2.1 Systemfunktioner

Systemet fungerade i stort enligt beställarens önskemål med några undantag. Aggregeringsrutinen gick inte att köra då någon av slutanvändarna hade dokument öppna, vilket ledde till att några aggregeringar blev mer tidsödande än planerat. Från ekonomiavdelningens sida fanns även en osäkerhet om när alla hade budgeterat klart. Någon form av kvittens hade, enligt den budgetansvarige varit önskvärd. Tillägg av nya kostnadsställen samt felaktig inmatning av slutanvändare ledde till en felräkning som upptäcktes vid aggregeringskörning. Det senare problemet är dock ett problem även vid manuell budgetering. Ekonomiavdelningen hade support som gjorde att ingen behövde sitta fast i systemet någon längre tid. I övrigt fungerade systemet som avsett.

6.2.2 Användningsfrekvens

För att observera användningsfrekvens undersöktes i vilken utsträckning de kostnadsställeansvariga utnyttjade den i nätverket lagrade informationen. Den budgetansvarige uppskattade att en majoritet sällan eller aldrig tittade på vare sig budget eller

utfall när budgetarbetet väl var klart. Knappt hälften utnyttjade systemet regelbundet för att följa upp den egna budgeten.

6.2.3 Orsaker och effekter

Den budgetansvarige tolkade den varierade användningsfrekvensen delvis som en *generationsfråga*. En annan faktor (eventuellt överlappande), som påverkade användningen negativt var slutanvändares ovana att utnyttja bildskärm som *informationsbärare* istället för papper. En faktor som påverkade användningen positivt var *nyttinsikt*. Ett exempel på detta var en slutanvändare i 60-årsåldern utan tidigare datorvana som kommit att utnyttja datorn och det lokala nätet för andra ändamål efter att budgeteringsprojektet genomförts. För slutanvändare, som hade erfarenhet av kalkylprogram upplevdes systemet som en möjlighet till *interaktiv simulering*, vilket medförde möjlighet till *självständig ekonomisk analys* som i sin tur gjorde budgetarbetet *snabbare* och gav det en *högre kvalitet*. Avdelningschefen på underhållsavdelningen, som är den avdelning som har flest kostnadsställen formulerar möjligheten till förändringar:

"Att budgetera manuellt är knöligt. Tänk Dig att man skall fördela 90.000 timmar på 80 kostnadsställen med papper och penna."

	A	B	C	D	E	F	G	H	I
1	LOGOTYPE		Kostnadsställe				Benämning		
2	BUDGETUNDERLAG								
3	Tryck TAB för att flytta mellan de inramade gula inmatningsfälten! Tryck Ctrl+Home för utgångsläge								
4	Konto	Kategori	Antal	Normtid	Sum tim	Övertid tim	Tot tim	Timlön kr	Sum lön tkr
5									
6		<i>Daggående</i>							
7	5011	Drift	99,0	1 808	178 992	0	178 992	91,08	16 303
8	5011	Underhåll	0,0	1 808	0	0	0	96,91	0
9	5011	Vikarier	*****	*****	0	0	0	90,00	0
10			99,0		178 992	0	178 992		16 303
11									
12		<i>Kont 2-skift</i>							
13	5011	Flenseri	0,0	1 704	0	0	0	122,15	0
14	5011	Utlastning/Hartsokeri	0,0	1 704	0	0	0	123,84	0
15	5011	Vikarier	*****	*****	0	0	0	120,00	0
16			0,0		0	0	0		0
17									
18		<i>3-skift</i>							
19	5011	Ordinarie	0,0	1 656	0	0	0	139,38	0
20	5011	Vikarier	*****	*****	0	0	0	137,00	0
21									
22									
23	5011	Totalt budget 1996	99,0		178 992	0	178 992		16 303

Figur 67 Exempel på inmatningsblad i budgetsystemet.

Chefen för efterbearbetningen, som budgeterade direkt vid datorn menade att budgetarbetet blev snabbare och roligare då man direkt kunde se effekterna av sin fördelning. Han uttrycker det: "Det är ingen nackdel att det är roligt." Han menade även att det fanns kostnadsställeansvariga som blev mer intresserade av datorer, genom att de var tvungna att arbeta med budgetsystemet.

En annan effekt av systemet var att medarbetare kunde kommunicera via nätverket. Flera av budgetposterna berörde flera avdelningar. Man kunde då granska budgetarna via nätverket. Underhållschefen reflekterade att budgeten blev mer officiell när den låg i nätverket och var tillgänglig för övriga berörda.

För ekonomiavdelningen gav budgetarbetet möjlighet till ett slags simuleringar, eftersom man kunde göra sammandrag kontinuerligt. Möjligheten att påverka parametrar som var kopplade till budgetarna gjorde simuleringen mer kraftfull. En nackdel med projektet var att man upplevde ett visst beroende av programmeringsexpertis. Ekonomiavdelningens kontakter med slutanvändarna var positiv med något undantag. Dessa undantag gällde slutanvändare utan datorvana.

6.2.4 Hantering

Slutanvändare kunde välja mellan att delegera inmatningen av budgeten eller budgetera själv. Då man budgeterade själv kunde detta ske med varierande grad av interaktivitet. I det mest interaktiva fallet kunde slutanvändaren utföra hela budgeteringsarbetet vid datorn och endast ta ut utskrifter för kontroll. I det minst interaktiva fallet gjorde den kostnadsställeansvarige hela budgeteringen med papper och penna en sekreterare sköta inmatningsarbetet vid datorn.

Flera slutanvändare noterade att *utskrift* av budgetblad och utfall var ett problem. Utskriften gjordes då man ville få översikt och föredrog papperskopior. Långsamma skrivare i nätverket och okunnighet om hur man skrev ut flera blad på en gång gjorde operationen tidsödande. Slutanvändare som utnyttjade systemet frekvent uppmärksammade möjligheten med att kunna skriva ut budgeteringsbladen allteftersom man budgeterade.

Budgetunderlaget var uppdelat på *flera kalkylblad*. För slutanvändare med liten datorvana ledde detta i vissa fall till svårigheter att använda systemet. För den budgetansvarige och vana kalkylprogramanvändare uppfattades det som en stor fördel att kunna samla de olika budgetuppgifterna på olika blad i en och samma arbetsbok.

6.2.5 Beställares deltagande i utvecklingsarbetet

Hantering av textfiler sköttes helt av beställaren med undantag av programmeringen av konvertering och integration (se ovan). Design av budgeteringsunderlaget skedde till allra största delen av beställaren. Undantag var de delar som behövdes för anpassning till programmeringen, vilket var en mycket liten del av arbetet och som inte berörde bladens utseende och funktion.

6.3 Diskussion

Undersökningen gäller utveckling av ett informationssystem i kalkylprogrammiljö där systemutvecklingen initierats, drivits och till stora delar utförts av de verksamhetsfö-

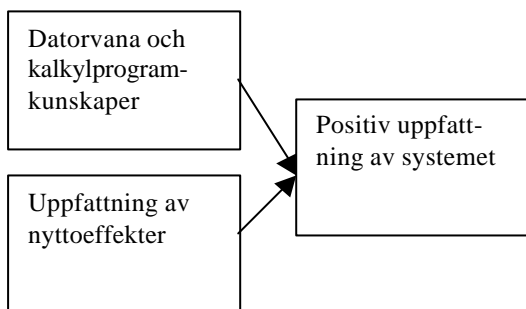
reträdare som var systemägare. Alternativ till ovan beskrivna tillvägagångssätt kan vara

- ingen utveckling, dvs fortsatt manuell budgetering
- traditionell systemutveckling i projektform med dataavdelningen eller externa konsulter som utförare
- inköp av standardsystem

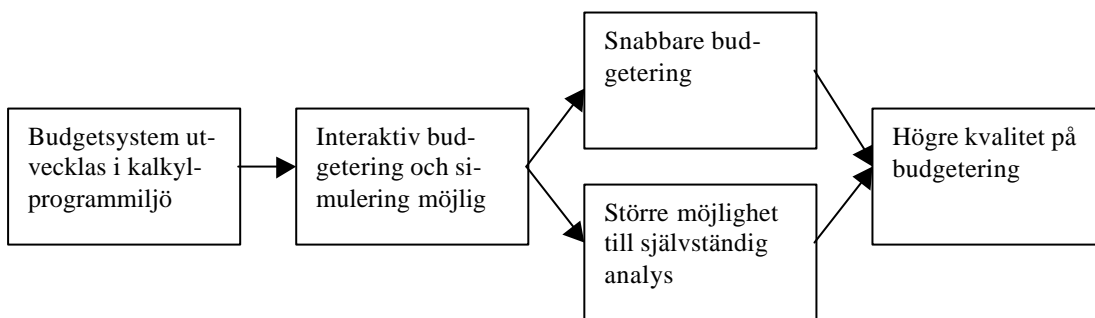
De centrala aktörerna är personalen på ekonomiavdelningen, som valde denna utvecklingsform istället för något av alternativen. Deras upplevelser av utvecklingsarbetet är intressant för denna avhandling som behandlar förutsättningar för och effekter av verksamhetsföreträdare som bedriver systemutveckling med kalkylprogram. Som en del av studierna av effekter behandlas även uppfattningar som systemets användare gett uttryck för. Diskussionen nedan, som handlar om hur resultatet kan tolkas, delas därför upp i aspekter rörande användning och utveckling, där användning handlar om hur systemets slutanvändare upplevt det utvecklade systemet och utveckling om förutsättningar för och effekter av verksamhetsföreträdare utveckling av en applikation.

6.3.1 Användning

Att slutanvändare inte utnyttjade systemet för att följa upp budgetar i avsedd omfattning hade såväl verksamhetsorienterade som (kalkyl)systemorienterade orsaker. En viktig verksamhetsorienterad orsak var att de centrala avdelningarna på företaget bedrev en verksamhet vars utgifter styrdes/styrs av produktionsbehov och inte av budget. För att ta ett exempel, om en maskin går sönder kan reparation eller nyinköp inte avgöras utifrån om budget kommer att överskridas eller ej. Om maskinen skulle bli stående skulle företaget i princip förlora sina intäkter. Av detta skäl har inte budgeten samma styrinstrumentförmåga som t.ex. i en avgiftsfinansierad verksamhet (som t.ex. ett universitet). (Kalkyl)systemorienterade orsaker till att systemet inte utnyttjats i avsedd omfattning var att en majoritet av de kostnadsställeansvariga inte hade någon större *datorvana* och upplevde det som onaturligt att ta del av information via bildskärm när man var van vid papper. Den del av slutanvändargruppen, som kom att utnyttja systemet hade större datorvana och/eller mer omfattande kalkylprogramkunskaper än övriga slutanvändare. Denna grupp kom att uppfatta systemet positivt. Minst ett undantag från regeln med datorvana kunde dock konstateras, vilket pekar på att en egenskap som kan kallas nyttoeffekt kan ha minst lika stort inflytande som datorvana och kunskap om kalkylprogramanvändning. När det gäller hanteringsaspekter uppfattades vissa funktioner som fördelar av de ovana och som nackdelar av de vana. *Interaktiviteten* i systemet var en avgörande faktor för positiv uppfattning. Förutsättningar och effekter sammanfattas grafiskt i Figur 68 och Figur 69 nedan.



Figur 68 Förutsättningar för positiv uppfattning av budgetsystemet.



Figur 69 Effekter av budgetsystemet ur användarnas perspektiv.

6.3.2 Utveckling

För att bygga ett budgetsystem av den typen som beskrivits ovan krävs olika kunskaper av olika *kunskapstyp*.

1. Kunskaper om budgetering, t.ex. förekommande budgeteringsmetoder
2. Kunskaper om företagets verksamhet ur ett budgeteringsperspektiv, t.ex. vilka kostnadsställen som finns och hur de är relaterade
3. Kunskaper om programvara för att utveckla systemet, inmatning, bearbetning, lagring, presentation och distribution³
4. Kunskaper om nätverk, för att skapa en kostnadsställestruktur med behörigheter i nätverket

1. Den första kunskapstypen är professionskunskap för ekonomer
2. Den andra kunskapstypen tillhör också de kunskaper som en ekonom på aktuellt företag förväntas ha
3. Den tredje kan man till viss del ha om man kan kalkylprogram
4. Den fjärde typen av kunskap måste vara centraliserad till någon enhet i en organisation, företaget sköter dataavdelningen denna typ av uppgifter

De första två kunskapstyperna hör till ekonomarbetsuppgifterna och den fjärde typen gör det inte. Den tredje typen av kunskap har traditionellt hört till IT-specialister, men har i takt med att utvecklingsverktyg blivit mer lättillgängliga varit möjlig för verk-

³ Enligt Langefors klassiska definition av informationssystem.

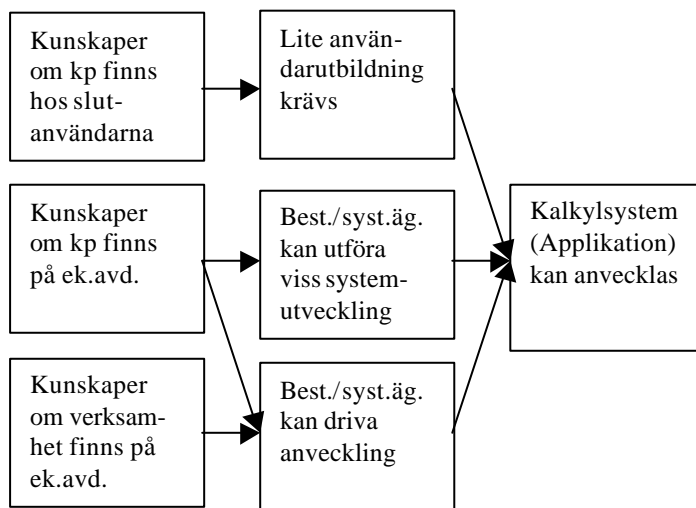
Applikation

samhetsföreträdare att förvärva till viss del. Eftersom kunskap om kalkylprogram fanns på ekonomiavdelningen låg det nära till hands att använda kalkylprogram för att kunna utföra systemutvecklingsuppgifter. Den del av utvecklingsarbetet som man inte utförde var programmeringsarbetet.

Att det budgetsystelet ledde till budgetering med högre kvalitet än manuell budgetering säger inte att det inte skulle blivit högre kvalitet på budgeteringen om systemet utvecklats traditionellt eller om man använt ett standardsystem. Undersökningen tar inte ställning till andra sätt att bygga budgetsystelet Undersökningen säger att det utvecklade systelet blev som det var tänkt, att kvaliteten ökade och att ett antal positiva bieffekter erhöles.

Sammanfattningsvis kan sägas att undersökningen visar (se även Figur 70 nedan)

- att kalkylprogram kan användas för att utveckla en *fleranvändarapplikation*
- att inkörningen av systelet kräver liten insats av *användarutbildning* eftersom många av användarna var bekanta med kalkylprogrammiljön
- att vissa väsentliga delar av väsentliga *utvecklingsaktiviteter* (t.ex. design av indata och utdata) kan utföras av beställare/systemägare
- att beställare/systemägare kan *driva ett utvecklingsarbete* (vad gäller tidsplanering, systemverifiering) utifrån arbetsuppgiftsperspektiv
- att *kunskaper i kalkylprogramanvändning* hos beställare/systemägare (och slutanvändare) påverkar möjligheterna att arbeta arbetsintegrerat med kalkylsystemutveckling



Figur 70 Slutsatser av undersökningen.

Ovanstående innebär att ett utvecklingsarbete kan initieras, drivas, delvis utföras, verifieras, delvis förvaltas och utvärderas av verksamhetsföreträdare utifrån systemägarrens/beställarens arbetsuppgiftsperspektiv.

DEL III

DEL III

Referensram

"Den struktur som valts här har ambitionen att vila på tre kriterier, *stabilitet, relateringsmöjlighet, och fokuseringspotential.*"
(Avdic 1999:140)

- *Syftet med detta kapitel är att diskutera hur en relevant referensram för studier av kalkylprogramanveckling (KPA) bör vara beskaffad.*

7 Referensmodell

Referensram och referensmodell ses här som åtskilda begrepp. Med referensram avses här en teoribakgrund till studieområdet KPA (och till avhandlingen) i form av värderingar och föreställningar bestående av kunskap relaterad till studieområdet. Referensmodellen är den form som referensramens kategorier är ordnade i. Den består av ett antal kategorier och relationer mellan dessa. Referensmodellen representerar vidare ett övergripande perspektiv på ett studieområde. Referensramen är referensmodellen fylld med innehåll. Samma referensmodell kan användas till flera studieområden.

Del III inleds, i detta kapitel, med en diskussion kring hur referensramen bör vara kategoriserad, dvs utformningen av referensmodellen. Därefter behandlas kategorierna i kapitel 8 till 15. I anslutning till respektive kategori redogörs för tidigare forskning inom studieområdet.

7.1 Strukturering av referensram

En viktig del av kunskapsutveckling är att strukturera den utvecklade kunskapen och relatera denna till annan relevant kunskap. Detta strukturerande bör lämpligen ordnas på ett sätt som hjälper såväl författaren som läsaren att orientera sig i den mängd av information som finns. Hur struktureringen skall göras är inte självklart. I den aktuella informationsmängden är en mängd strukturer möjliga. Någon av dessa väljs, medvetet eller omedvetet, och styr därefter författares och läsares väg in i studieområdet. Olika strukturer ger olika perspektiv på studieområdet. Nedan görs ett försök att göra strukturen, eller referensmodellen som den kallas här, genomskinlig, dvs att fokusera på

strukturen som sådan i stället för den kunskap som den syftar till att strukturera och relatera.

7.2 Kriterier för och kategorier i referensramen

Hur struktureras då referensramen till ett studieområde utan etablerade traditioner? Hur struktureras referensramen till ett studieområde som oavbrutet förändras och omprövas såväl till innehåll som till form? Den struktur som valts här¹ har ambitionen att vila på tre kriterier, *stabilitet*, *relateringsmöjlighet*, och *fokuseringspotential*. Med *stabilitet* menas att referensmodellen strävar mot att ha ett drag av grundläggande tidlöshet. Referensramens centrala begrepp skall vara av den generella karaktären att förändringar inom studieområdet inte skall behöva leda till förändringar i referensmodellen. Att enbart sträva efter stabilitet kan dock leda till en meningslös referensram, då de allra mest stabila begreppen samtidigt är de mest generella, t.ex. fenomen, objekt och händelse. Referensramen bör anknyta till studieområdet mer än så. Läsaren måste uppfatta referensramen som meningsfull, och denna meningsfullhet uppstår när läsaren kan relatera uppfattad kunskap med befintlig kunskap. *Relateringsaspekten* innebär i detta specifika fall att anveckling skall kunna relateras till fenomen som dels är kända av målgruppen och som dels är påtagligt relaterade till anveckling. Det fenomen som främst används för relatering är traditionell systemutveckling (TSU). Motivet till detta val är att anveckling här framställs som ett specialfall av systemutveckling och därför lämpligen kan relateras till någon etablerad modell av systemutveckling, vilket i detta fall är vad som här kallas traditionell systemutveckling.² De valda kategorierna i referensramen syftar alltså även till att framhäva likheter och skillnader mellan anveckling och TSU. Då de båda fenomenen (anveckling och TSU) skulle kunna ses som uttryck för två systemutvecklingsparadigm³, syftar kategorierna till att lyfta fram kvalitativa skillnader mellan dessa båda paradigm. Det tredje kriteriet *fokuseringspotential*, avser möjligheterna att fokusera på väsentliga problem, möjligheterna att analysera studieområdet på ett meningsfullt sätt och att framhäva det specifika med anveckling.

Avhandlingens studieområde är *kalkylprogramanveckling* (KPA), vilket generellt sett kan betraktas som en aktivitet utförd av människor. Valet av kategorier utgår av detta skäl från en så generell beskrivning som möjligt av mänsklig aktivitet. Det språkliga uttrycket *Någon gör något i något sammanhang i något syfte* kan sägas utgöra en sådan allmän beskrivning. I delprojekt 1 (licentiatavhandlingsprojektet) användes en sådan generisk allmän språklig sats för att strukturera referensramen. Ur nämnda sats skulle följande kategorier kunna härledas:

- *Någon* → **Aktör**. Den centrala aktören i KPA är anvecklaren själv.
- *gör* → **Process**. Processen i KPA är själva utvecklingsarbetet.
- *något* → **Produkt**. Produkten i KPA är kalkylsystemet.

¹ En referensmodellmodell!

² Ytterligare motiv till valet av TSU redovisas nedan.

³ Inte i den relativt strikta form som Kuhn (1962) använde begreppet, men ungefär som Nurminen (1988) använder det då han talar om paradigmatiska egenskaper för att skilja tre perspektiv på informationssystem.

- *i något sammanhang* → **Förutsättningar**. Förutsättningar påverkar anvecklarens situation. En av de centrala förutsättningarna för anvecklingsprocessen, är kalkylprogrammet, dvs ett av de materiella verktygen för anveckling.
- *i något syfte* → **Intention**. Anvecklaren är en person som utför självständigt arbete mot mål.

Utifrån satsen ovan skulle olika kategorier kunna fokuseras. Om t.ex. *produkt* skulle fokuseras så skulle kalkylsystemet vara utgångspunkten och de övriga kategorierna relateras till produkten kalkylsystem. Om *intentionen* med anveckling fokuserades skulle övriga kategorier relateras till anvecklares intentioner med anveckling. Som titeln på denna avhandling antyder så är det process och aktör som är i fokus, vilket innebär att området är tänkt att belysas ur ett perspektiv som fokuserar dessa kategorier. Då syftet är att relatera förutsättningar och effekter bör även detta speglas av det valda perspektivet. Särskilt viktiga förutsättningar är kunskap och verktyg, vilket behandlats i kapitel 1. Ett antal modeller skulle kunna komma ifråga som kandidat till referensmodell för kalkylprogramanveckling. Bland dessa kan nämnas Nurminens (1988) HIS-modell, Activity Theory (Engeström 1991) och Action Workflow (Action Technologies 1993). Då de två senare inte explicit behandlar det för KPA centrala kunskapsbegreppet faller dessa. HIS-modellen är visserligen tillämplig och förklarar KPA till stora delar (se kap 8.4) men kan upplevas som väl allmän för att återspegla det (kronologiska) handlingsperspektiv som KPA lämpar sig att betraktas ur.

För att återspegla såväl huvudkategorier som det kronologiska förhållandet i begrepps-paret förutsättningar – effekter, samt för att uppfylla ovan beskrivna krav på stabilitet och relateringsmöjlighet, kommer *den praktikgeneriska modellen* (Goldkuhl & Röstlinger 1998) att användas som referensmodell för att strukturera studieområdet. Ytterligare ett argument för att utnyttja den praktikgeneriska modellen är att den har stora likheter med den grundläggande modell som användes i projektets del ett (licentiatavhandlingen) och kan ses som en fördjupning och specificering av den ursprungliga modellen (*Någon gör...*).

7.3 Den praktikgeneriska modellen

Goldkuhl & Röstlinger (1998) har formulerat *den praktikgeneriska modellen* i syfte att skapa en grund för verksamhetsutveckling och teoriutveckling. Praktik definieras enligt

"En praktik innebär att någon/några gör något för någon/några, utifrån uppdrag från någon/några, och att detta görande (handlande) baseras på värden, regler och kunskaper och kompetenser som är etablerade och som successivt kan förändras." (Goldkuhl & Röstlinger 1996)⁴

Modellen är aktörs- och handlingsorienterad och betonar den intentionella aspekten med praktik. Kunskap framställs som starkt integrerat i en praktik. Kunskap kommer till uttryck i handlingar utförda av aktörer i praktiken.

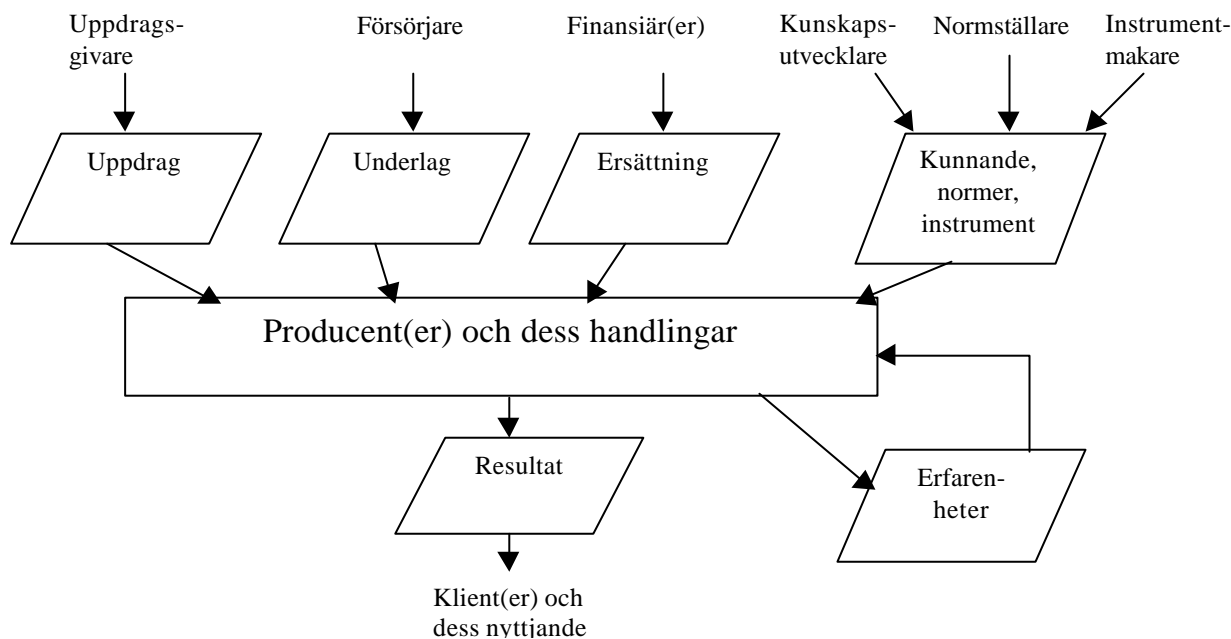
⁴ För diskussion av innebörden i begreppen kunskap och kompetens, se kapitel 13.

Den praktikgeneriska⁵ modellen beskriver aktörer som utför handlingar utifrån vissa förutsättningar och med vissa resultat (Goldkuhl & Röstlinger 1997:5). Förutsättningarna är input till praktiken, vilken genererar resultat och erfarenheter. Erfarenheterna kan återgå till praktiken. En praktik innebär framställning av något, ofta genom att använda sig av något. Den aktör/de aktörer som utför handlingarna kallas i modellen för producent(er). En anvecklare använder sig av förutsättningar i form av t.ex. av kalkylprogram, indata, kunnande av olika slag för att kunna skapa ett kalkylsystem som genererar information av något slag. Även andra förutsättningar påverkar möjligheterna att åstadkomma ett önskat resultat. Den praktikgeneriska modellen identifierar grundläggande förutsättningar i form av

- uppdrag
- underlag
- ersättning
- kunnande, normer och instrument

Förutsättningarna är uttryck för intentionella handlingar utförda av andra aktörer.

Förhållandet mellan kategorierna visas i Figur 71 nedan.



Figur 71 Den praktikgeneriska modellen (Goldkuhl & Röstlinger 1998:7).

Nedan ges en kort redogörelse för de centrala begreppen i den praktikgeneriska modellen med exempel från anvecklingsområdet. Först ges en beskrivning av modellen som den presenteras av Goldkuhl & Röstlinger (1998) och efter det diskuteras viss modifiering av modellen i syfte att anpassa den till studieområdet kalkylprogramanveckling och syftet med denna avhandling. Beskrivningen av modellen börjar med förutsättningarna för att sedan fortsätta med producent, resultat och slutligen erfarenheter.

⁵ Generisk i meningen grundläggande.

Referensmodell

En praktik baseras på *uppdrag*. Uppdraget kommer från *uppdragsgivare*. En handläggare kan ha sin chef som uppdragsgivare. En företagare kan ha en kund som uppdragsgivare. En anvecklare kan ha sig själv som uppdragsgivare (se vidare nedan). Den praktikgeneriska modellen utgår från att det alltid finns minst en uppdragsgivare. Uppdraget är ett uttryck för uppdragsgivarens *intentioner*.

Normalt utför aktörer handlingar med någon form av *underlag* som input. Underlaget kommer från en *försörjare*. Underlag kan för en anvecklare vara indata i form av t.ex. produktionsdata eller prisuppgifter. Försörjare är de personer som producerar underlaget. Produktionsdata kan t.ex. komma från andra system där de verifierats av någon med behörighet till detta. Prisuppgifter kan komma från inköpschefen i företaget. En försörjare kan vara producent i en annan praktik. Denna praktiks resultat kan då bli underlag i andra praktiker. Underlag kan passera flera led.

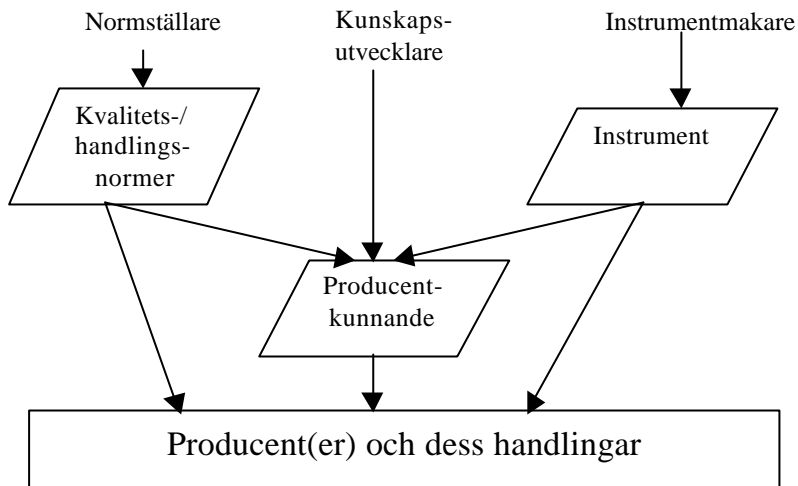
En *finansiär* betalar på något sätt den *ersättning*, som normalt utgår till producenten i en praktik. I vissa verksamheter kommer ersättningen direkt eller indirekt från verksamhetens externa klienter. Privat företagsamhet är exempel på detta. Om en handläggare på ekonomiavdelningen på ett företag utvecklar kalkylsystem får han sin lön från företaget, som i sin tur lever på sina kunders betalning.

Förutsättningarna kunnande, normer och instrument är till viss del överlappande. Med *kunnande* avses den kunskap och den kompetens som behövs för att utföra praktikens handlingar. En anvecklare behöver olika typer av kunnande. Exempel på kunnande som en ekonomianvecklare kan behöva ha är kunskap om hantering av kalkylprogrammet, kunskap om nätverk och lagringsmedier, kunskap om yrket t.ex. redovisning samt kunskap om den specifika verksamheten t.ex. om produktgrupper. Den aktör som producerar och förmedlar kunnande och kompetens kan kallas *kunskapsutvecklare*.

Normer formuleras av *normställare*. ”Kvalitets-/handlingsnormer reglerar producenternas handlingsutrymme.” (Goldkuhl & Röstlinger 1997:8) Kvalitetsnormer syftar till att avgöra huruvida handlingars resultat är bra eller mindre bra. Handlingsnormer talar om vad som får och vad som inte får göras.

Instrument konstrueras av *instrumentmakare*. Instrument är hjälpmedel eller verktyg av olika slag som producenten behöver i sin praktik. Instrument kan vara materiella (t.ex. utrustning) eller immateriella (t.ex. metoder). Exempel på anveklarinstrument kan vara datorer, kalkylprogram eller metoder för datastrukturering. I fortsättningen kommer benämningen *verktyg* att användas synonymt med instrument.

Användare och utvecklare



Figur 72 Relationer mellan normer, kunnande och verktyg (Goldkuhl & Röstlinger 1997:11).

De förutsättningar som beskrivits ovan kan sägas vara interna eller externa. Med *interna förutsättningar* menas förutsättningar, som finns i förhållande till producenten/anvecklaren. Kunnande är en intern förutsättning. Som en del i producentens kunnande ingår normer och verktyg. De är dock i viss utsträckning oberoende av producenten, vilket motiverar att de förs fram som egna förutsättningar. Kunnande, normer och verktyg är följaktligen delvis överlappande. *Externa förutsättningar* finns följaktligen i någon mening mer oberoende av den specifika producenten/anvecklaren. Förhållandet mellan normer, kunnande och verktyg visas med Figur 72 ovan. Figuren syftar till att visa att ”producentkunnande inbegriper åtminstone delar av normer och immateriella verktyg (metoder)”.

Den centrala aktören i en praktik är *producenten*. Producenten utför, under givna förutsättningar, handlingar i sin praktik, för att skapa resultat, i enlighet med det uppfattade uppdraget. Flera producenter kan ingå i en praktik om de har samma förutsättningar och följaktligen strävar efter att generera samma typ av resultat.

En praktik genererar *resultat* i form av varor eller tjänster. Den som tillgodosgör sig resultatet kallas enligt den praktikgeneriska modellen för *klient*. Resultatet som en anvecklare producerar beror på i vilken verksamhet praktiken bedrivs. Resultat kan å ena sidan delas upp i varor och tjänster och å andra sidan i materiella, informativa och upplevelsemässiga produkter.

En exemplifiering på resultat och klient i ett anvecklingssammanhang kan vara en handläggare som arbetar med reklamationsuppföljning. Om handläggaren förser marknadschefen med reklamationsstatistik är detta ett resultat och marknadschefen är klient. Resultatet är då en informationsvara. Det som i handläggarens praktik är ett resultat kan i marknadschefens praktik vara ett underlag.

I en praktik utförs handlingar som leder till nytt kunnande i form av *erfarenheter*. Nya erfarenheter skapar nya möjligheter att utföra praktik. En anvecklare lär sig t.ex. nya möjligheter att lösa problem med kalkylprogram.

7.4 Praktikgenerisk modell för anvecklare

En producent styrs av sin praktiks uppdrag. Praktiker kan ha olika komplexitet. En enkel praktik utmärks av att producenten har väldefinierade uppdragsgivare, försörjare, finansiärer, klienter, normställare, kunskapsproducenter och verktygsmakare. Ju mer komplex verksamheten blir desto mer komplext kan det bli att identifiera de olika aktörerna bakom beteckningarna.

En faktor som styr komplexiteten i modellens tillämpning är *uppdragets karaktär*. Två ytterlighetsfall, procedurbaserad och målbaserad praktik (Sprague & Watson 1996:7) (se även kap 12), kan demonstrera detta. Om uppdraget innebär att förväntningarna på resultatet är precisa och innefattar direktiv för hur resultat skall uppnås så kan praktiken sägas vara *procedurbaserad*. En mycket procedurbaserad praktik innebär att det är preciserat vilket underlag som skall användas, vilka normer som gäller, vilket kunnande som krävs samt även vilka verktyg som skall användas. Ett exempel på procedurbaserad praktik är löpande band-arbete där den enskilde producenten har att följa givna regler i sitt arbete. Denna typ av aktivitet förknippas normalt inte med anveckling.

Om uppdraget är mindre preciserat, dvs om underlag, kunskaper, verktyg och eventuellt normer är mindre preciserade så skulle detta kunna kallas för en *målbaserad* praktik. Även målbaserade praktiker kan vara av olika typ. Å ena sidan (*operativa-målpraktiker*) kan praktiker ha klart uttryckta, väl specificerade mål av typen ”Avkastningen skall vara 10% under året” eller ”Enligt budget skall vi producera 1000 ton/dag”. Å andra sidan kan praktiker (*operativa-målpraktiker*) ha mindre kvantitativa, specificerade, mer kvalitativa, strategiska, mål av typen ”Ökad kundnytta”, ”Hög kvalitet” eller ”Kostnadseffektiv produktion”.⁶ Även om en praktik styrs av precisa mål av den typ som nämns ovan och detta påverkar verksamheten på ett mycket påtagligt sätt, så kan det finnas andra väsentliga mål som anvecklare och andra måste ta hänsyn till. Målen kan vara outtalade, inofficiella, olagliga, motsägelsefulla och icke-rationella, men de kan finnas där och påverka verksamheten. (Goldkuhl & Röstlinger 1988:86) Verksamhetsetik och yrkesetik som beskrivs i det inledande kapitlet är exempel på ospecificerade men verksamma mål som styr anvecklarens verksamhet. Det är inte heller säkert att ett väl specificerat operativt mål är tillämpligt på alla situationer en anvecklare ställs inför.

När det gäller målbaserade praktiker blir begreppet *ansvar* centralt. Ju mindre preciserade förutsättningar och ju mindre preciserat mål, desto större ansvar vilar på anvecklaren. En anvecklare har ansvar för en uppgift, avdelning eller funktion, vilket innebär att konsekvenserna av handlingsresultat i praktiken kan komma att drabba anvecklaren personligen på ett eller annat sätt. Det personliga ansvaret för en avgränsad uppgift är intimt kopplat till utförandet av handlingen. Man kan föreställa sig att ansvaret är nivåindelad över en formaliserbarhetsvariabel tillsammans med övriga kategorier. Inne-

⁶ Denna typ av praktik är besläktad med vad som skulle kunna kallas värdebaserad praktik. Mål- och värdebaserad praktik diskuteras även i kapitel 14.

bördan av resonemanget blir att ju mindre formaliserat uppdraget är desto mer ansvar vilar på producenten.

I en målbaserad praktik avgör producenten själv vilket underlag som krävs och hur det skall införskaffas. Även kunskapsbehov fastställs av producenten. I vissa fall kan även normer formuleras av producenten. Hela praktiken kan sägas vara mindre formaliserad i en målbaserad praktik jämfört med en procedurbaserad. Allra minst formaliserad är den i en strategiska-mål-baserad praktik. Ett exempel är en ekonomichef som i sin verksamhet har att fullfölja de intentioner som verksamhetsledningen meddelar honom genom att framföra vilka effekter som önskas i verksamheten i form av t.ex. ”goda möjligheter till uppföljning av produktion och administration”.

Målbaserad praktik kan leda till oformaliserat ansvar, oformaliserade uppdrag, ospecificerade underlag, normer på en hög abstraktionsnivå, generella verktyg samt generell och speciellt kunnande (se Tabell 7 nedan). De målbaserade praktiker är av särskilt intresse eftersom de medför beslutssituationer där verktygstillverkning i form av kalkylsystem kan bli aktuell. Utformningen av verktyget styr sedan vilket underlag som krävs för praktiken. Anvecklaren bestämmer också hur normer, kunskap och verktyg skall kombineras. I denna avhandling är inte procedurbaserade och välstrukturerade praktiker av intresse eftersom verktygstillverkning inte tillhör vanligheterna i dessa praktiker. Intentioner med kalkylsystem behandlas mer ingående i kapitel 12.

Tabell 7 Skillnader mellan procedurbaserad och målbaserad praktik

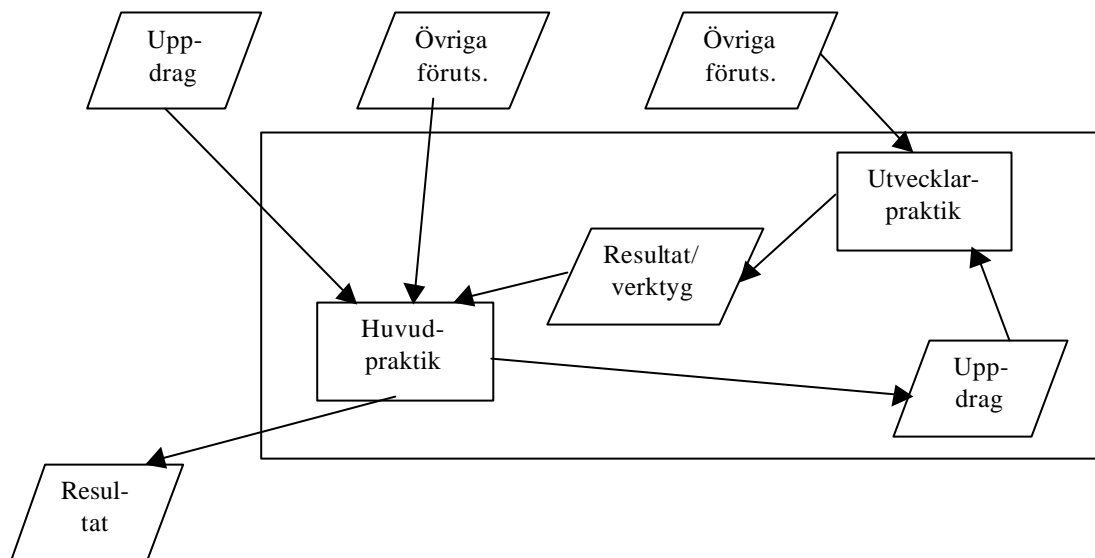
	Procedurbaserad praktik	Operativa-målbaserad praktik	Strategiska-målbaserad praktik
Uppdrag	Precist formulerat	Formulerat som preciserade mål	Formulerat som strategiska mål
Ansvar	Preciserat	Kan vara preciserat eller opreciserat	Oformaliserat eller på hög abstraktionsnivå
Underlag	Preciserat	Ospecificerat, kan behöva skapas eller hämtas	Ospecificerat, kan behöva skapas eller hämtas
Kunnande	Visst preciserat kunnande krävs	Generellt och speciellt kunnande krävs eller kan behöva inhämtas	Generellt och speciellt kunnande krävs eller kan behöva inhämtas
Normer	Väldefinierade	Kan vara preciserade eller mindre preciserade	Kan behöva formuleras eller på hög abstraktionsnivå
Verktyg	Väldefinierade	Generella och speciella verktyg krävs eller kan behöva utvecklas	Generella och speciella verktyg krävs eller kan behöva utvecklas
Praktik	Välstrukturerad	Ej definierad på förhand	Ej definierad på förhand
Resultat	Definieras i uppdraget	Preciserat	Beror på tolkning av uppdraget

Referensmodell

En frågeställning som aktualiseras då struktureringsgraden i en praktik diskuteras är om en person, som bedriver en komplex praktik kan sägas utöva en eller flera praktiker. Ett sätt att se på denna frågeställning är att se en persons verksamhet som en praktik oavsett hur komplex den är. Om detta är fallet så är aktörerna bakom förutsättningarna andra personer än producenten. Själva praktiken kan i detta fall vara komplex och innefatta många typer av aktiviteter. Ett annat sätt att se det är att identifiera flera praktiker. Man kan tala om en kedja av praktiker. (Goldkuhl & Röstlinger 1997:9) Om man identifierar flera praktiker i en persons verksamhet är frågan hur man avgränsar en viss praktik och vad som i så fall styr avgränsningen. Ett avgränsningskriterium skulle kunna vara att en praktik definieras med en persons/en arbetsgrupps arbetsuppgifter, som nämnts ovan. Ett annat kriterium skulle kunna vara att en praktik kan sägas tillhöra ett *kunskapsområde*. I det första fallet (en praktik per person) skulle det, ur ett anvecklingsperspektiv, inte gå att skilja på de arbetsuppgifter som producenten utför som systemutvecklare från övriga uppgifter. Detta skulle innebära att den praktikgenetiska modellen inte gav samma analytiska stöd som om det var möjligt att särskilja de systemutvecklande uppgifterna. I det andra fallet, med en praktik per kunskapsområde, och eventuellt med flera praktiker per person, skulle olika delar av en producents verksamhet kunna skiljas från varandra. Detta skulle kunna leda till utveckling av kunskap om hur olika typer av kunskap, normer och verktyg påverkar (olika) praktiker i olika situationer. Av nämnda skäl väljs den senare avgränsningen (en praktik per kunskapsområde).

En praktik utövas följaktligen inom ett *kunskapsområde*, en domän. Ett exempel på ett kunskapsområde skulle kunna vara kalkylsystemutveckling, ett annat externredovisning och ett tredje kartongmaskintillverkning. Att det finns specifika kunskaper som kan vara svåra att entydigt placera i det ena eller andra området behöver inte diskvalificera det meningsfulla med indelningen. Om praktikern/producenten verkar inom flera kunskapsområden utövas flera praktiker. En anvecklare kännetecknas av att han utövar minst två praktiker. Anvecklaren har en huvudpraktik och en utvecklarpraktik i vilken han utvecklar a-system, vilka är resultat i utvecklarpraktiken och som fungerar verktyg i huvudpraktiken.

Ett exempel på innebörden i att anvecklaren verkar i olika praktiker är om en ekonomichef tillverkar ett budgeteringssystem för den verksamhet där han arbetar. Dels har han sin ekonomichefspraktik där han förväntas generera resultat i form av budgetar, bokslut, efterkalkyler etc. Om ekonomichefen behöver tillverka kalkylsystem för att utföra efterkalkyler innebär detta att han måste tillverka ett verktyg. Verktøget tillverkar han i en annan praktik som systemutvecklare. Systemutvecklingspraktiken innehåller andra normer, annat kunnande och andra verktyg än ekonomichefspraktiken (se Figur 73 nedan). Om producenten verkar inom



Figur 73 Anvecklarpraktiken innehåller i sin tur minst två praktiker.

flera praktiker samtidigt är det en praktik som är *huvudpraktiken*. För ekonomichefen är det praktiken som ekonomichef som är huvudpraktiken. Ytterligare en praktik skulle kunna identifieras, nämligen verksamhetspraktiken, vilket diskuteras vidare i kapitel 14. I denna avhandling är det dock anvecklarpraktiken, bestående av en utvecklar(del)praktik och en yrkes(del)praktik, som är i fokus.

Utifrån ovanstående resonemang är det möjligt för en anvecklare att som person vara sin egen klient, uppdragsgivare, försörjare, normställare, kunskapsproducent och verktygsmakare, fast i olika (del)praktiker. Om ekonomichefen tillverkar ett kalkylsystem för efterkalkyl så kan uppdraget betraktas som om det kommer från ekonomichefspraktiken till utvecklarpraktiken. Ekonomichefen är i ekonomichefspraktiken klient till utvecklarpraktiken. I utvecklarpraktiken skapas ett resultat i form av ett verktyg. Verktøget fungerar som en förutsättning i ekonomichefspraktiken. Men kan även återgå som erfarenheter i form av kunnande eller verktyg i utvecklarpraktiken.

Erfarenheter kan i en praktik innebära att det skapas nya verktyg, normer och nytt kunnande. För en anvecklare är det kontinuerliga lärandet en viktig aspekt som vävs samman med utveckling och användning (Avdic 1995a:181). Lärandet kan resultera i nytt kunnande om kalkylsystemutveckling. Även nya normer kan utvecklas i form av insikter om vad som är bra och dåligt (kvalitetsnormer) eller vad som får eller inte får göras (handlingsnormer). Verktøg som en anvecklare tillverkar kan användas inte bara i producentens praktik utan även i andra praktiker.

Referensmodell

Om man relaterar de inledningsvis beskrivna grundläggande termerna aktör-produkt-process-omgivning-intention med den praktikgeneriska modellen så kan följande kopplingar göras:

- Produkt – Resultat
- Process – Handlingar i en praktik
- Aktör – Producent
- Omgivning – Förutsättningar
- Intention – En aspekt av uppdrag

Resultatet är i en anvecklares utvecklarpraktik definitionsmässigt ett informationssystem, ett anvecklat system eller ett *a-system*. Om det är fråga om KPA är resultatet ett kalkylsystem. I anvecklaren huvudpraktik fungerar kalkylsystemet som ett verktyg med vars hjälp anvecklaren producerar resultat som är kopplat till uppdraget i huvudpraktiken. Uppdrag och resultat har följaktligen ett intimt samband. I uppdraget ligger en förväntning om ett visst resultat. Uppdragsgivarens intention är ett uttryck för vilken effekt som förväntas av resultatet. När det gäller systemutveckling innehåller resultatet normalt ett informationssystem. Aspekter kring utvecklarpraktikens produkt och därmed uppdragsgivarens intentioner behandlas vidare i kapitel 8.

Handlingar i syfte att utveckla informationssystem i en anvecklares utvecklarpraktik kan definieras som en *process*. Det är inte fråga om slumpvisa handlingar eller specifika handlingar som kan utföras i godtycklig ordning. Handlingarna syftar till att konstruera ett informationssystem och präglas av detta ändamål. Då det i denna avhandling finns en ambition att relatera anveckling till TSU och denna ofta beskrivs som en process motiverar detta användandet av ordet process. Aspekter kring praktiken som sådan behandlas vidare i kapitel 9 och 10.

I en anvecklares praktik finns alltid flera aktörsroller. Dels är anvecklaren producent i sin huvudpraktik, men även i sin utvecklarpraktik. Anvecklaren kan ur ett systemutvecklingsperspektiv vara utvecklare, in användare, ut användare, formulerare, beställare, systemförvaltare etc. Anvecklaren roller som systemutvecklare behandlas vidare i kapitel 11.

I den praktikgeneriska modellen kan förutsättningarna delas upp i norm-kunskapsrelaterade och övriga. Bland de övriga förutsättningarna (uppdrag, underlag och finansiering) behandlas inte finansiering i denna avhandling. Som tidigare diskuteras påverkas praktiken i hög grad av uppdragets karaktär. Procedurbaserat uppdrag ger strukturerad praktik medan målbaserat uppdrag ger större utrymme för frihetsgrader vad gäller praktikens utövande. Underlag för en anvecklares praktik innehåller ofta information, som fungerar som indata till kalkylsystemet. En grov indelning kan göras i intern och extern information där den interna informationen finns inom organisationen och den externa utanför. Denna förutsättning behandlas i viss utsträckning i kapitel 10. Ytterligare förutsättningar av extern karaktär som är av betydelse för anvecklares praktik är aspekter som här samlas under samlingsnamnet organisatoriska aspekter, och som omfattar management, organisationskultur, lärande, utbildning, innova-

tion, ekonomi, kommunikation, systemarkitektur. Även dessa aspekter behandlas i viss utsträckning i kapitel 10.

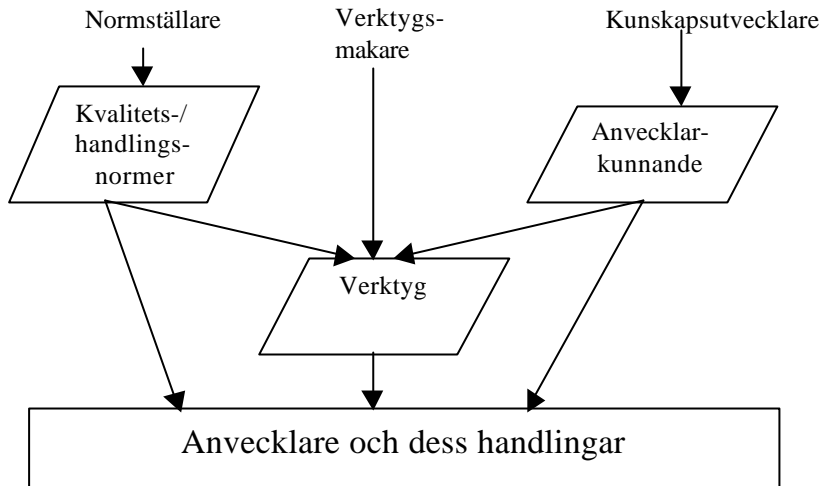
Ytterligare en förutsättning som kan sägas vara betydelsefull för möjligheten att producera resultat är faktorer och omständigheter som inverkar på aktörens situation är den utveckling som sker inom ett specifikt område och som styr kunnande, normer och verktyg utformning i olika avseenden. Denna utveckling är inom IT-området relativt snabb. Produktutveckling skapar möjligheter och vice versa i en takt som gör IT-området ytterst dynamiskt, vilket präglar de flesta verksamheter. Exempel på detta är utvecklingen av Internet, som skapat nya kommunikationskanaler. De norm-kunskapsrelaterade förutsättningarna kan i ett givet ögonblick ses som en ögonblicksbild av nämnda utveckling. Även övriga förutsättningar påverkas av den ständigt pågående utvecklingen. I det dynamiska IT-området har det ett intresse att spekulera över vad som kommer att ske härnäst. Ett sätt att göra detta är att se bakåt för att identifiera trender och identifiera områden där dessa trender gör sig gällande. Trender inom IT-området behandlades i kapitel 1.

Förutsättningarna, normer, kunskap och verktyg har även de sina speciella karaktäristika när det gäller anvecklarens utvecklarpraktik. Kvalitetsnormer för anvecklare (i utvecklarpraktiken) sammanfaller i viss utsträckning med systemutvecklarens kvalitetsnormer. Det framstår normalt som önskvärt att system skall vara användbara, vilket ofta leder till krav på användbarhet, systemflexibilitet, förändringsbarhet etc. Handlingsnormer kan regleras av myndigheter eller företagsledning. Ett visst mått av kunskaper om utveckling av informationssystem är det som utmärker anvecklaren. Kombinationen av dessa kunskaper med kunskaper om verksamheten är en grundläggande förutsättning för anveckling. Verktyg gör det möjligt för anvecklaren att utveckla kalkylsystem. Det viktigaste verktyget i kalkylprogramanvecklarens utvecklarpraktik är kalkylprogrammet, men andra, såväl materiella som immateriella verktyg kan användas. Resultatet i anvecklarens utvecklarpraktik, kalkylsystemet blir sedan verktyg i anvecklarens huvudpraktik. Aspekter kring verktyg behandlas vidare i kapitel 15.

Mellan kunnande, normer och verktyg finns, som nämnts ovan, en slags överlappning (se Figur 74 nedan). Vissa former av kunnande kan formaliseras och därmed också externaliseras⁷. Användning av verktyg kan här ses som ett sätt att överta och/eller utvidga producentens kunnande. Externaliserad kunskap och externaliserade normer finns följaktligen i någon mening oberoende av den specifika producenten/anvecklaren. Ju mer kunskap (och normer) som kan externaliseras och implementeras i verktyg, ju mer oberoende blir producenten i huvudpraktiken av IT-specialisters kunskaper i utvecklarpraktiken.

⁷ I meningen ”ges en yttre form”.

Referensmodell



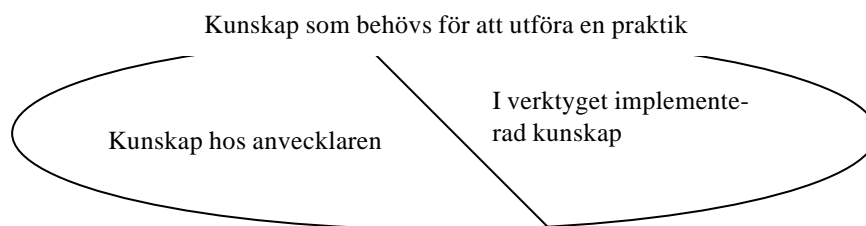
Figur 74 Relationer mellan normer, kunskande och verktyg i en anveklares praktik (Modifiering av Goldkuhl & Röstlinger 1997:11).

Normer och kunskaper ges en yttre form i t.ex. ett kalkylprogram. Det finns gott om såväl kvalitets- som handlingsnormer i ett kalkylprogram. Fördefinierade mallar, diagram, guider, defaultvärden, etc. är en form av kvalitetsnormer. Handlingsnormer kan yttra sig på olika sätt och vara mer eller mindre uppenbara eller genomskinliga. Ett exempel är att division med noll inte tillåts, (vilket också visas med ett felmeddelande). En rad juridiska, matematiska och fysiska begränsningar (t.ex. kalkylbladets storlek) kan alltså styra anveklarens handlingar och därmed praktik. Då anveklaren utvecklar sina system kan normer av olika slag läggas in i systemet. Att begränsa antalet valbara parametrar i ett simuleringssystem är exempel på detta. Att optimera värden utifrån specificerade variabelintervall är ett annat exempel. Normer kan externaliseras mer indirekt också. Om en handläggare gör ett kalkylsystem för att beräkna optimal pallstorlek utifrån orderstorlek och produktdimensioner kan det vara ett sätt att realisera miljömässiga och ekonomiska mål, som innebär att man inte skall slösa med resurser därför att de är ändliga och därför att de kostar pengar. Här kan det finnas skäl att skilja på handlingsnormer. Dels finns handlingsnormer som är önskvärda att följa (företagspolicy, operativa mål, juridiska lagar, etiska normer), dels handlingsnormer som man är tvungen att acceptera av tekniska eller andra skäl (t.ex. prestanda, naturlagar) men, som man strävar efter att övervinna om man kan för att uppnå den andra typen av handlingsnormer.

När det gäller kalkylprogram som verktyg så finns såväl normer som kunskande inbyggt i verktyget. Anveklaren får hjälp med att multiplicera matriser genom matrisbehandlingsfunktioner och normer om presentation styr utskrifter av pivottabeller. Sett ur detta perspektiv finns kunskande och normer inbyggt även i andra verktyg som hammare, systemutvecklingsmetoder och grävmaskiner. Kunskandet och normerna finns ju även hos anveklaren. Kunskandet och normerna fördelas mellan anveklaren och verktygsmakarna (se Figur 75 nedan). När det gäller kalkylprogram kan det hävdas att utvecklingen inom IT-området leder till att verktyget innehåller mer och mer av kunskande och normer. Om kunskande klassas som olika formaliserbart och verktyg i någon mening kan betecknas som ett slags formaliserat kunskande, så avbördas aktören

Användare och utvecklare

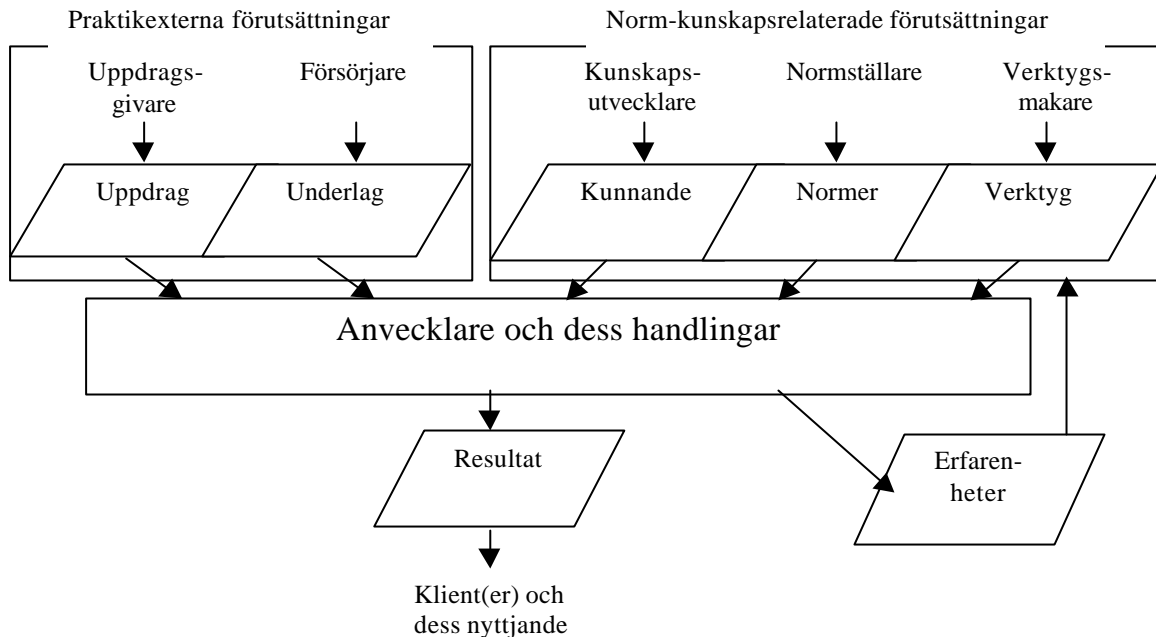
från vissa formaliserbara kunskaper och kan ägna sig åt de uppgifter som (än så länge) är svårformaliserbara.



Figur 75 Fördelning av kunskap mellan producent och verktyg.

När det gäller normerna finns också en koppling till verktyget. Verktygets begränsningar påverkar normerna. Kvalitetsnormer kan implementeras i verktyget med hjälp av guider, hjälpsystem, mallar etc. Handlingsnormer kan finnas inbyggda i ett verktyg på så vis att verktyget inte tillåter anvecklaren att göra vad som helst. I ett kalkylprogram kan t.ex. storleken på en databas inte överskrida antalet rader på ett kalkylblad. Ju mindre begränsningar och ju mer (välmotiverad) hjälp som finns i verktyget desto mer generellt blir verktyget vilket innebär att anvecklaren får ökat stöd (mer kunskap och mer kvalitetsnormer) och större handlingsutrymme (mindre handlingsnormer). Ökad verktygsgeneralitet innebär alltså ökat kunnande och mindre normer i verktyget. I Figur 76 nedan visas en modifierad version av den praktikgeneriska modellen. I denna modell definieras erfarenhet som något som återförs till praktiken som kunskap, normer och verktyg. Denna återföring är en följd av lärandeaspekterna i de sammanflätade aktiviteter som anvecklaren utför. Anvecklaren har tillgång till ytterligare norm-kunskapsrelaterade förutsättningar när nästa uppdrag kommer från huvudpraktiken. Exempel på sådana kan vara egendefinierade funktioner och mallar. Ytterligare två skillnader mot ursprungsmodellen är dels att ersättningsmodulen ej är representerad samt att kunnande, normer och verktyg representeras med egna romboider. Den senare förändringen syftar till att framhäva dessa förutsättnings betydelse som åtskiljare av olika praktiker. Förhållandet mellan de tre förutsättningarna tolkas också annorlunda än i ursprungsmodellen (se Figur 74 sid 151).

Referensmodell



Figur 76 Modifiering av den praktikgeneriska modellen (modifiering av Goldkuhl & Röstlinger 1997:7).

7.5 Om referensramen

Då studieområdets referensram med hjälp av referensmodellen skall beskrivas mer detaljerat i resten av *Del III*, inleds beskrivningen med kategorin informationssystem eftersom det är själva objektet för praktiken och referenser tillbaka på detta objekt förekommer i alla de andra kategorierna. Man kan också se detta som ett precedensanalytiskt angreppssätt på så vis att man börjar med resultatet och går sedan bakåt till praktiken (med praktikern/anvecklaren och processen) för att sedan behandla förutsättningarna för praktiken. Information från vissa referenser hamnar inte självklart i den ena eller andra kategorin. Ett exempel på det är forskning om anveckling/EUC. Denna behandlar olika aspekter av den praktikgeneriska modellen, men i vissa fall har det bedömts som väsentligast att presentera denna samlat och i några av dessa fall har uppgifter befunnits mer lämpliga att presentera under respektive kategori. Ett annat exempel på svårplacerad information är användbarhet. Diskussionen om användbarhet har placerats under kategorin verktyg (se kap 15). Hur andra gränfall har hanterats har varierat från fall till fall.

Information i avhandlingen som behövs för att formulera en referensram till studieområdet kalkylprogramanveckling har inte samlats genom användande av en i forskarsamhället väletablerad ämnesindelning. Insamlingen har inte heller skett utifrån specifika signum i bibliotekssystem, utifrån specifika sidor i fackpress eller utifrån etablerade sökbegrepp på Internet. Kalkylprogramanveckling är inte väletablerat vare sig som fenomen eller som begrepp. Det område som ligger närmast till hands är möjligen *End User Computing* (EUC). Inte heller *End User Computing* är dock ett heltäckande begrepp utan konkurrenter (se kap 10). Avsaknaden av en etablerad referensram att

Användare och utvecklare

positionera sig i, har för- och nackdelar. En nackdel är att det kan vara mindre enkelt att samla relevant information och en fördel är att det finns utrymme för att formulera begrepp och relationer utifrån egna ställningstaganden.

"By information systems we mean here:

A system of information sets needed for decision and signalling in a larger system (of which it is a subsystem) containing subsystems for

collecting

storing

processing

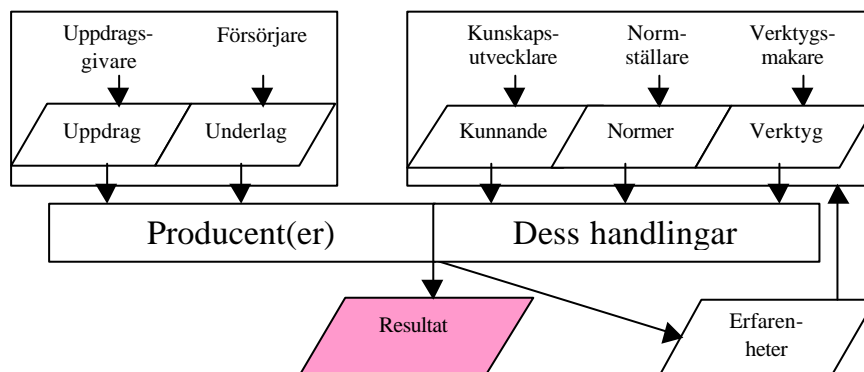
distributing

of information sets."

(Langefors 1966:143)

- *Detta kapitel syftar till att analysera informationssystembegreppet, som är resultatet i en systemutvecklingspraktik. Speciellt behandlas resultaten i anvecklarens utvecklarpraktik, a-system och kalkylsystem.*

8 Informationssystem



Det resultat som genereras av en systemutvecklingsprocess, det som utvecklaren utvecklar och det som användaren använder kan benämnas *informationssystem*. Informationssystem är en generell benämningen på den produkt kring vilket IT-området fokuserar.¹ Ett informationssystem består ur ett dataperspektiv av program och data, vilka representerar en dynamisk och en statisk aspekt av det som informationssystemet berör. Ett bankomatsystem innehåller t.ex. data om konton och bankkunder och det innehåller program som utför rutiner som berör dessa data. Då data tolkas av människor kallas det information. Informationssystem är följaktligen till för människor. Data är till för att representera information. Då det talas om informationssystem, är det system för människor som avses. För människor som använder informationen som produceras av informationssystemet är det normalt informationen som är intressant snarare än hur den är representerad i form av data. Detta är ett uttryck för ett informations-

¹ Även om andra förekommer, t.ex. IT-artefakt (Löwgren & Stolterman 1998).

perspektiv på informationssystem. Normalt finns det ett sammanhang och ett syfte bakom ett informationssystem, vilket är det som intresserar de aktörer som avses dra nytta av informationssystemet. För att systemet skall producera avsedd information på ett ändamålsenligt sätt vidtar systemutvecklare åtgärder avseende lagring, program, gränssnitt, utrustning etc. Det intressanta med informationssystemet är att det uppfyller sitt syfte. För bankomatsystemet gäller t.ex. att bankkunderna enkelt och säkert kan ta ut pengar på olika platser på olika tider.

En utgångspunkt för denna avhandling är att utvecklade system av typen kalkylsystem är informationssystem i den bemärkelse som skisseras ovan, med den skillnaden att de utvecklats av utvecklare. Av detta skäl är det intressant att skärskåda begreppet informationssystem, vilket sker i detta avsnitt.

8.1 Begreppet informationssystem

Informationssystem är ett av de mest centrala begreppen inom ämnesområdet ADB. Det är ett begrepp som finns med i de flesta ADB-böckers index. Definitionerna är många och ofta ganska lika, men vid närmare granskning kan vissa skillnader noteras. Enligt Sundgren (1992:27) kan informationssystem definieras utifrån *syfte* och *ingående funktioner*.

Ett informationssystemets *syfte*

"...är att understödja problemlösning och/eller beslutsfattande inom ett visst område, problemområdet eller intresse området. Närmare bestämt går denna uppgift ut på att

- tillhandahålla relevant och aktuell information kring sakförhållanden och skeenden inom intresseområdet;
- understödja relevanta bearbetningar av informationen" (Sundgren 1992:27).

Enligt Sundgren (1992:34) kan ett informationssystem vara helt informellt och således enbart baseras på manuell informationshantering. Eftersom alltigenom manuella informationssystem inte är av primärt intresse i denna avhandling, bortses i fortsättningen från dessa. För att undvika förväxling mellan *manuella informationssystem* och *informationssystem med automatiska inslag* används ibland uttrycket *datasystem* (Goldkuhl 1993:12).

En aspekt som har att göra med att informationssystem kan vara manuella eller mer eller mindre automatiserade är att även de manuella delar som krävs för att ett informationssystem skall uppfylla sitt syfte kan anses tillhöra informationssystemet. Såväl Andersen (1994:24ff) som Sundgren (1992:27ff) hävdar att så är fallet. Sådana manuella delar kan vara:

- instruktioner för när en bearbetning skall utföras
- kontrollberäkning av värden som skall matas in i informationssystemet
- (icke-formaliserade) regler för vad inmatade värden betyder
- rutiner för distribution av rapporter från systemet
- bedömningar om behandlingen av informationen varit korrekt.

Genom att definiera vad ett informationssystem är, drar man en gräns mellan vad som (enligt definitionen) kan anses vara ett informationssystem och vad som inte (enligt definitionen) kan anses vara det. Helt okomplicerad är dock inte gränsdragningen mellan ett informationssystem och den verksamhet som systemet stödjer eller är en del av. Goldkuhl (1993:14) menar att "Behandlingen av information utanför datasystemet kännetecknas av varierande grad av formalisering och rutinisering", vilket innebär att det enligt det synsättet finns funktioner som inte kan anses tillhöra informationssystemet. Rimligt är ju också att det dras en gräns någonstans. Alternativet är att betrakta hela organisationen (eller hela världen) som ett informationssystem. Gränsen mellan informationssystemet och den omgivande verksamheten kan alltså inte betraktas som självklar utan beror på hur informationssystembegreppet definieras och hur informationssystemets syfte definieras. Funktioner som hamnar inom syftet med informationssystemet kan anses tillhöra detsamma. Exempel på sådana funktioner nämns ovan. I denna avhandling räknas även icke-automatiserade delar som krävs för att systemet skall uppfylla sitt syfte till informationssystemet. Detta är en konsekvens av det arbetsintegrerade perspektivet. Genom att användningen av kalkylprogram integrerat flyter in i det ordinarie linjearbetet ter det sig för anvecklaren onaturligt att skilja ut delar som tillhörande systemet eller inte på tekniska grundvalar (för exempel se kap 3 och 4). Den naturliga utgångspunkten för informationssystemets syfte är de arbetsuppgifter som användaren har.

Den klassiska definitionen av ett informationssystemets *funktioner*, som haft inflytande på många efterföljande definitioner, är den som Börje Langefors presenterade 1966:

"By information systems we mean here:

A system of information sets needed for decision and signalling in a larger system (of which it is a subsystem) containing subsystems for

collecting

storing

processing

distributing

of information sets."

(Langefors 1966:143).

Langefors definierade alltså ett informationssystem som ett system, vars uppgifter är att samla in, lagra, behandla och distribuera informationsmängder. Langefors grundsyn benämns ibland informationsteoretisk (Bansler 1990:104ff) eller systemteoretisk (Nurminen 1988:21ff). Langefors perspektiv var ett uttryck för en syn på informationssystem som möjliga (och ibland önskvärda) att analysera och sönderdela (hierarkiskt). Informationssystem ses enligt den systemteoretiska synen i någon mån som ett objekt i sig, delvis till skillnad från en syn som ser informationssystemet som ett uttryck för mänskliga intentioner. Langefors hade inte ett handlingsperspektiv på informationssystem.² Delar av Langefors syn på informationssystem återkommer i andra definitioner, t.ex. Berg & Hultman (1985:18), Olle m.fl. (1991:20) och Andersen

² Detta är inte att uppfatta som kritik av Langefors. Han var pionjär vad gäller att placera användaren i fokus och hans arbete måste ses i ljuset av den tekniska utvecklingen. På senare tid har Langefors till viss del modifierat sina teorier (Langefors 1993).

(1994:15). I de två senare definitionerna berörs systemets omgivning mer än vad som var fallet med Langefors uppfattning. Till en del kan det hänföras till att man med tiden mer och mer uppfattat informationssystem som en del av en större verksamhet där det skall spela en roll som skall överensstämja med verksamhetens mål. Informationssystemet har ur verksamhetens perspektiv inget egenvärde, en tolkning som man inte helt kunde utesluta från den systemteoretiska definitionen.

Andersens (1994:15) definition av informationssystembegreppet får tjänstgöra som utgångspunkt för denna avhandlings sammanfattande definition av begreppet *informationssystem*. Andersen sammanfattar sin uppfattning om vad ett informationssystem är så här:

"Ett informationssystem är en mänsklig konstruktion."

"Ett informationssystem måste vara knutet till en viss arbetsuppgift."

"Ett informationssystem förmedlar information från vissa personer till vissa personer."

"Ett informationssystem tar emot information av olika slag."

"Ett informationssystem utför olika typer av informationsbehandling."

"Informationsbehandlingen i ett informationssystem kan vara både manuell och maskinell."

(Andersen 1994:12)

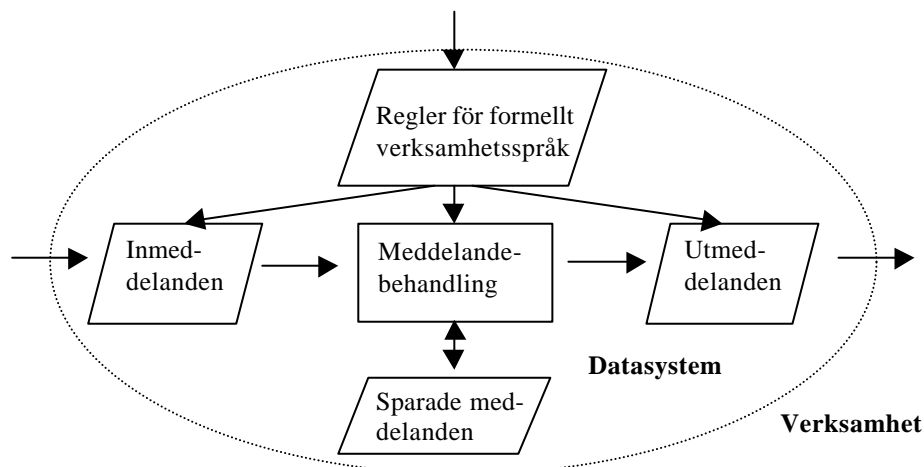
Andersen sammanfattar informationssystemets syfte:

"Berättigandet och betydelsen av ett informationssystem kan bara förstås i förhållande till en verksamhet. Ett informationssystem har ingen mening i sig. Det existerar för att tjäna en verksamhet. Insamling, bearbetning, lagring, överföring och presentation av information kan inte utföras på ett vettigt sätt om man inte känner till verksamheten och dess mål och uppgifter." (Andersen 1994:22)

De centrala begreppen för beskrivning av informationssystemets funktioner är *insamling*, *bearbetning*, *lagring*, *överföring* och *presentation*. Med *insamling* avses här de rutiner i informationssystemet som ha till uppgift att formalisera informationen och göra den tillgänglig för databearbetning. Exempel på insamlingsfunktioner är programrutiner för inmatning av uppgifter via ett formulär, t.ex. när man matar in kod och belopp i en bankomat. Även icke-automatiserade rutiner räknas till insamlingsfunktionerna. Med *bearbetning* avses de funktioner i informationssystemet som "...kan utföra transformation av informationen i kunskapsbasen på olika sätt, t.ex. genom matematiska beräkningar, mängdoperationer, logiska operationer och statistisk analys" (Sundgren 1992:29). Enligt Andersen (1994:15) skulle bearbetningsfunktionen kunna delas upp i "...aritmetiska beräkningar och sammanställning av information från olika källor". Om man definierar ett informationssystem som funktioner som faller inom syftet med informationssystemet, skulle en manuell bedömning av utdata kunna anses vara en bearbetning inom ramen för informationssystemet. Med *lagring* avses funktioner för att bevara informationen kortare eller längre tid (Sundgren 1992:29). "För att utföra sina informationsskapande och kommunikativa uppgifter behöver datasystem vanligtvis *spara* (lagra) information över tiden" (Goldkuhl 1993:15). Överföring och presentation definieras något olika i de anförda källorna. Med *överföring* avses funktioner för

att överföra "...information från en person eller grupp av personer till en annan person eller grupp" (Andersen 1991:22). I själva begreppet information finns en överförande aspekt. Information uppstår då data tolkas. Detta innebär att informationssystemets utdata är till för att informera någon. "Ett datasystem³...relaterar olika grupper av människor till varandra" (Goldkuhl 1993:15). Begreppet överföring måste här ses i ljuset av hur gränsen mellan informationssystemet och den omgivande verksamheten definieras. Om gränsen dras helt nära den automatiserade delen så kan enanvändarsystem, som har samma användare för att hantera både inmatning och utmatning, inte betraktas som informationssystem. Om däremot syftet med informationssystemet är vägledande för hur gränsen, mellan vad i verksamheten som tillhör respektive inte tillhör informationssystemet, skall dras så är normalt även personliga enanvändarsystem att betrakta som informationssystem. Med *presentation* avses funktioner för hur information visas för användaren eller informationsmottagaren. Samma information kan presenteras på flera olika sätt. Olika sätt att presentera kan resultera i olika måluppfyllelse för ett visst system. I kalkylprogramsammanhang spelar ofta presentationsaspekter stor roll.⁴

Då enbart informationssystemets datahanterande funktioner berörs kan man tala om ett datasystem. Datasystemet består, som nämnts ovan, av funktioner för inmatning, bearbetning, lagring och utmatning. För att konstruera datasystemet måste det ske en formalisering av regler för registrering, behandling, lagring och utmatning av data (se Figur 77 nedan).



Figur 77 Datasystem i verksamhet (Goldkuhl 1993:16).

Ett informationssystem består följaktligen dels av data och dels av regler för hur data skall hanteras. Formaliseringen av regler i ett informationssystem kan ske på olika sätt. Ett sätt är programmering ett annat är formelinmatning i ett kalkylblad. Inmatning av data till de databehandlande funktionerna är användning av systemet. (I de fall som data hämtas från databaser kan användning innebära att starta systemet.) I informationssystem av typ bankomat eller andra traditionellt utvecklade system (se nedan) är de olika modulerna (i Figur 77 ovan) normalt åtskilda. Programmet lagras i ett minnesutrymme, data i ett annat. I programmet finns olika moduler som hanterar inmatning,

³ Används av Goldkuhl synonymt med uttrycket *datorbaserat informationssystem*.

⁴ Se t.ex. Olles **pris- och indexutveckling** Figur 52 sid 96.

bearbetning och utmatning. I ett kalkylsystem, t.ex. **beräkning av arkantal per pall** (se kap 3.3), kan alla funktionerna vara integrerade i kalkylbladet. Data matas in i vissa celler där även lagring sker. Bearbetning sker i celler som omgäende visar utdata. Denna integration av informationssystemfunktioner utgör ett kännetecken för kalkylsystem.⁵

Synen på informationssystem kan, enligt ett sätt att indela, sägas vara mer systemteoretiskt orienterad som i de definitioner som beskrivs ovan, eller mer handlingsorienterad. En handlingsorienterad syn på informationssystem, formuleras av Goldkuhl (Goldkuhl 1993:15).

"Ett datasystem

- ingår i och är avsett att stödja en omgivande verksamhet
- realiserar (på basis av mänskligt formulerade instruktioner) olika språkliga handlingar, som är informationsskapande och kommunikativa, dvs datasystemet utför arbetsuppgifter i verksamheten
- bygger på och inbegriper språkliga utsagor som bygger på en verksamhetsvokabulär - innehåller därmed ett formellt verksamhetsspråk dvs det är fördefinierat, standardiserat och avgränsat
- relaterar alltid olika grupper av människor till varandra."

Denna syn på informationssystem betonar de språkliga och kommunikativa funktionerna. I ett informationssystem har man "bakat in" och också "fryst" ett verksamhetsspråk, som varit aktuellt vid tiden för utvecklingen av systemet. "Computer-based information systems are rule-governed formal language action. This language action consists of inferencing and communication." (Goldkuhl 1984:4)

Betonandet av informationssystemets språkliga och kommunikativa karaktär är ett uttryck för en förändring i synsätt (jämfört med de inledande definitionerna) vad gäller uppfattningen av informationssystem som fenomen. "Information systems possesses the potential to support authentic communicative interaction between organizational actors in such a way as to inspire criticism, to facilitate learning, and to raise the overall level of information systems." (Lyytinen & Hirschheim 1988) Definitionerna (de systemteoretiska och de handlingsorienterade) utesluter inte varandra utan uppmärksammar olika aspekter av informationssystem. Det handlingsorienterade synsättet fokuserar informationssystemet som en integrerad del av en verksamhet, medan det systemteoretiska synsättet mer fokuserar själva informationssystemet som ett objekt som avbildar ett objektsystem.(Goldkuhl 1996:3) Det handlingsorienterade synsättet är relevant i beaktande av att informationsanvändare själva har möjlighet att utföra eller åtminstone ha ett tongivande inflytande över utformningen av informationssystemet. När systemutveckling bedrivs som en i det ordinarie linjearbetet integrerad aktivitet för att uppnå vissa verksamhetsmål, framstår den handlingsorienterade definitionen som mer relevant och verksam än den systemteoretiska. I det ordinarie linjearbetet används informationssystem för att uppfylla mål och uttrycka intentioner i form av det verksamhetsspråk som är grunden för lösande av arbetsuppgifter. Dessa handlingar uttrycker intentioner hos sändaren. I kapitel 3 och 4 kan samtliga exempel sägas vara

⁵ Integrationen gäller dock inte kalkylsystem av typen *Applikation*, se nedan, där programkod normalt lagras separat.

uttryck för denna handlingsorienterade syn. Den arbetsintegrerade karaktären både på användningen av kalkylsystem och utvecklingen av dem gör att denna verksamhet av anvecklaren själv inte bedöms som kvalitativt annorlunda än andra handlingar som anvecklaren utför i sin verksamhet. Den handlingsorienterade definitionen av informationssystem skiljer sig alltså från den traditionella Langeforsdefinitionen i sin syn på informationssystemets roll. Den traditionella synen betraktar i högre grad ett informationssystem som ett fenomen i sig självt, en nedbrytbar artefakt med ett eget berättigande, medan den handlingsorienterade definitionen placerar informationssystemet i en kontext av människor som utför uppgifter mot mål. Den handlingsorienterade synen på informationssystem är naturligt applicerbar på kalkylsystem.

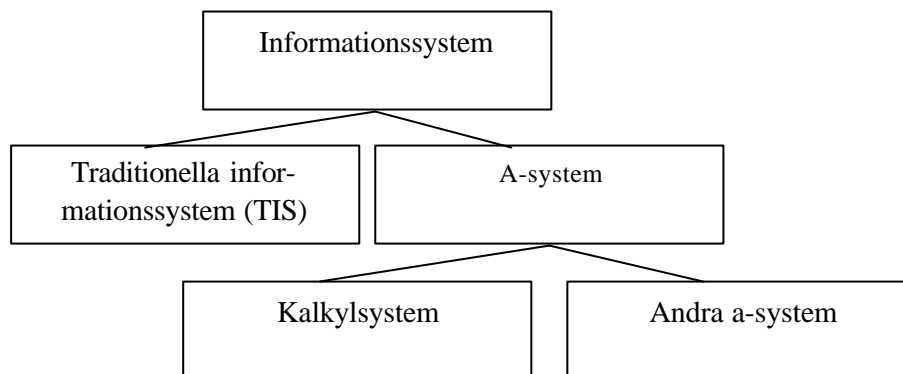
Den handlingsorienterade synen är influerad av talaktsteori och socialfenomenologi. Enligt *talaktsteorin* (Searle 1969) gör vi något när vi säger något. Ett yttrande är en handling. En grundläggande del i talaktsteorin består i att skärskåda yttranden. Ett yttrande kan betyda något helt annat än vad det kanske bokstavligen betyder. Ironi är ett uttryck för ett sådant yttrande. Detta innebär att tolkning är något väsentligt i talaktsteorin (och i informationsbehandling). Ett yttrande kan tolkas olika beroende på egenskaper hos såväl sändare som mottagare. Yttranden uttrycks med språk. Informationssystem är 'fryst' tal i form av ett formaliserat språk. Följaktligen kan information i informationssystem betraktas som handlingar med sändare och mottagare. Dessa handlingar uttrycker intentioner hos sändaren. Informationssystem kan uppfattas som "...a social, linguistic entity..." och "...'social systems only technically implemented'. They are formal linguistic systems for communication between people which support their actions." (Goldkuhl & Lyytinen 1982:2) Enligt *socialfenomenologin* (Berger & Luckman 1969) kan informationssystem ses som institutionaliserade mänskliga språkliga handlingar. Om språket betraktas som medium för information, så ger socialfenomenologin ett perspektiv på information där språket framställs som en produkt av objektiverade tankar och känslor som kan ta sig olika uttryck. Språklig objektivering av tankar och känslor kan leda till institutioner i form av informationssystem. Eventuellt kan dessa informationssystem reifieras, dvs komma att uppfattas som fenomen som inte är mänskliga konstruktioner.

Sammanfattning av definition på informationssystem:

- Ett informationssystems *syfte* är att understödja problemlösning och/eller beslutsfattande inom ett visst område
 - Ett informationssystem har *funktioner* för insamling, bearbetning, lagring, överföring och presentation av information.
 - Ett informationssystem skall *stödja en verksamhet*
 - Ett informationssystem har *såväl automatiska som manuella delar*.
 - Ett informationssystem *används* i vid bemärkelse av *flera människor*
- Dessutom bidrar den handlingsorienterade definitionen med några för KPA viktiga aspekter:
- Ett informationssystem innehåller (delar av) ett verksamhetsspråk
 - Ett informationssystem uttrycker institutionaliserade kommunikativa handlingar mellan människor.

8.2 Traditionella informationssystem (TIS) och anvecklade informationssystem (a-system)

Definitionen av informationssystem, i kapitel 8.1 ovan, kan tillämpas på alla informationssystem. Men eftersom definitionen är övergripande finns det olika sätt att dela in mängden informationssystem. En indelning skulle kunna göras i traditionella informationssystem (TIS) och anvecklade informationssystem (a-system)⁶ (se Figur 78 nedan). Ett TIS är utvecklat med traditionell systemutveckling (TSU) och ett a-system är anvecklat (se kap 10).



Figur 78 Kalkylsystem är en typ av a-system som är en typ av informationssystem.

Under den tid då systemutveckling var förbehållen systemutvecklingsspecialister, var informationssystem en produkt av en specialistdriven process, som här kallas traditionell systemutveckling (TSU) (se kap 9). Tillsvi vidare definieras ett TIS som ett informationssystem som utvecklats

- som en följd av någon form av formell analys
- under projektliknande former
- med metodstöd
- av specialister
- med utvecklingsverktyg avsedda för specialister
- och överlämnats till beställare och användare
- och förvaltas av speciell personal.

Skälen till att just dessa egenskaper lyfts fram diskuteras närmare i kapitel 9.

Ett a-system är

1. utvecklat av, eller i nära samarbete med, personer som arbetar med eller har nytta av a-systemet eller som har kunskap om och ansvar för (delar av) den verksamhet som a-systemet skall betjäna.
2. utvecklat parallellt med att anvecklaren sköter sina arbetsuppgifter
3. ofta förändringsbart för användaren
4. ofta ett mindre system
5. ofta utvecklade för att stödja en persons eller ett fåtal personers arbetsuppgifter

⁶ Skulle även kunna kallas EUC- system.

6. utvecklade med verktyg som är avsedda för anvecklare, t.ex. kalkylprogram eller databashanteringsprogram

Exemplen i kapitel 3 och 4 uppfyller samtliga punkter (1 till 6) ovan, medan exemplet i kapitel 6 uppfyller alla utom 4 och 5.

Även om ovanstående punkter är uppfyllda är det inte säkert att det utvecklade systemet uppfyller utvecklarens förväntningar. Lally (1995:3) definierar lyckade användarutvecklade system⁷ som lämpliga för användarutveckling, logiskt korrekta och användbara för sitt ändamål. Skillnaderna mellan ett TIS och ett a-system är främst att ett TIS är utvecklat av en IT-specialist medan ett a-system är utvecklat (anvecklat) av en utvecklare.

8.3 Kalkylsystem

Kalkylsystem är en typ av a-system, som är en typ av informationssystem (se Figur 78 ovan). ”Ett kalkylsystem är ett informationssystem, som är utvecklat med hjälp av kalkylprogram och som fungerar integrerat i kalkylprogrammet. Med detta menas att kalkylsystemet består dels av kalkylprogrammet självt dels av en modul som utvecklats för något ändamål. Verktøget som använts för att skapa kalkylsystemet ingår alltså i själva kalkylsystemet.” (Avdic 1995a:7).

I ett kalkylsystem kan informationssystemets delar integreras på ett (eller flera) kalkylblad. Data kan registreras direkt i kalkylbladet där det också kan lagras. Formler och funktioner, som bearbetar data kan registreras direkt i kalkylbladet, vilket också kan fungera som utdatamedium. Integrationen av informationssystemets delar ger förutsättningar för interaktivitet, där en snabb växling mellan användning och regelformulering är möjlig. I och med att kalkylprogrammet ingår i kalkylsystemet, kan kalkylsystemet utnyttja färdigformaliserade, generella bearbetningsfunktioner som finns tillgängliga i kalkylprogrammet. I *Del II, Empiri* finns exempel på kalkylsystem.

De personer, som utvecklat kalkylsystemet, kan vara användare, handlingsansvariga eller engagerade i linjeorganisationen på annat sätt. Begreppet kalkylsystem som sådant bygger dels på en syn att kalkylsystem är intimt sammanflätade med den verksamhet i vilken de finns, dels på en syn att man kan indela informationssystem efter den utvecklingsmiljö, i vilken de är konstruerade. Man skulle följaktligen kunna tänka sig att i det sammanhanget tala om desktopsystem, CAD-system, statistiksystem, ordbehandlingssystem osv.

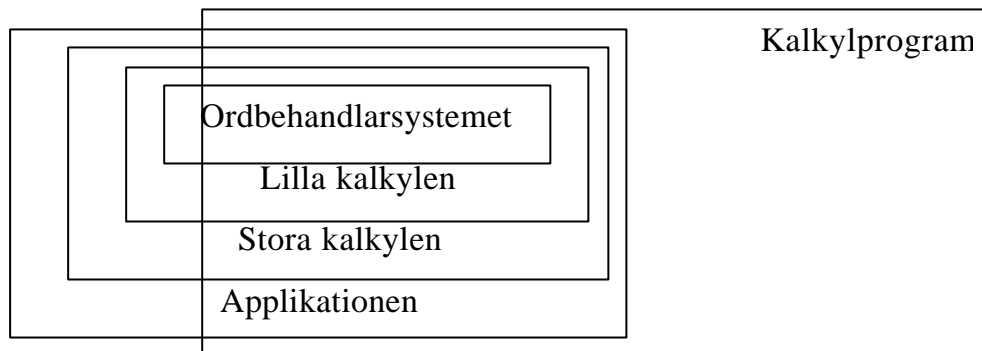
Kalkylsystem kan variera i storlek och komplexitet från mycket enkla system, utvecklade på några minuter till fleranvändarsystem utvecklade under flera år. En indelningsgrund för analys av olika kalkylsystemtyper är utvecklarens verktygskunskaper. Nedanstående indelning i kalkylsystemtyper (se Figur 79 nedan) bygger på detta: (Avdic 1995a:123)⁸

- Ordbehandlarsystemet

⁷ Userdeveloped systems.

⁸ I Avdic (1995a:119ff) finns flera utförliga exempel på de olika typerna.

- Lilla kalkylsystemet
- Stora kalkylsystemet
- Applikationen



Figur 79 Kalkylsystemtyper och kalkylprogram (Avdic 1995a:185).

Alla kalkylsystem har det gemensamt att själva kalkylprogrammet är en del av systemet. Detta medför att det finns en inneboende möjlighet till utveckling eller förändring av systemet. Här finns en skillnad mot system som kan implementeras som s.k. run-time moduler. Ett sådant system kan laddas in i datorn och köras som de är. Det innehåller dock inte någon utvecklingspotential för anvecklaren av den typ som utmärker kalkylsystemet.

Ordbehandlarsystemet

Ett kalkylsystem av typen *Ordbehandlarsystemet* är närmast att betrakta som en form av strukturerad ordbehandling (se Avdic 1995a:121f). Anvecklaren lagrar data på ett kalkylblad på ett ordnat sätt av något skäl, ofta för att kunna göra en utskrift. Ordbehandlarsystemet innehåller alltså inte någon form av bearbetning. Denna mycket vanliga användning av kalkylprogram behandlas sällan uttryckligen i litteratur om kalkylprogramanvändning som systemutveckling (t.ex. Kaiser & Oppelland 1990, Nelson 1989, Panko 1988). Enligt Langefors' (1966:143) definition på informationssystem är det inte självklart om detta definitionsmässigt kan betraktas som ett informationssystem. För anvecklaren är dock denna skillnad inte uppenbar. En avgörande faktor är även var man drar gränsen för informationssystemet. Om man innefattar manuella rutiner före inmatning av data och efter uttag av data kan de flesta Ordbehandlarsystem räknas som informationssystem även med Langefors' definition.

Ett Ordbehandlarsystem kan enkelt omvandlas till Lilla Kalkylsystemet. Ordbehandlarsystemet kan för anvecklaren vara en inkörsport till mer avancerad utveckling. Ett exempel på ett Ordbehandlarsystem som omvandlats till Lilla Kalkylsystemet är Olles **budgetering av exploateringsföretag** som redovisas i kapitel 4.

Lilla kalkylsystemet

Lilla Kalkylsystemet innehåller enkla formler⁹ och SUMMA-funktioner i begränsad utsträckning. Ett system av typen Lilla kalkylsystemet innehåller inte sällan diagram. I

⁹ Av typen $=A1+B1$ eller $=\text{Intäkt}-\text{Kostnad}$.

kapitel 3 och 4 finns flera exempel på Lilla Kalkylsystemet, t.ex. Ismos **fördelning av arkning** och Annettes **sammanställning av en personalenkät**.

Stora kalkylsystemet

Stora Kalkylsystemet innehåller mer komplicerade formler och funktioner än Lilla Kalkylsystemet. Systemet kan vara fördelat på flera kalkylblad. Användarens förkunskaper måste innefatta viss förståelse av datorns funktioner och uppbyggnad (t.ex. rörande enheter, kataloger och filer), vilket inte är lika nödvändigt för de första två typerna.

Då kalkylsystemutveckling omnämns i artiklar eller böcker (t.ex. Ledell 1993, Westin 1993, Carlsson 1993, Storey & Chan 1994, Isakovitz, Shocken & Lucas Jr 1995) är det i allmänhet Stora kalkylsystemet som behandlas. Exempel på Stora Kalkylsystemet är t.ex. Bosses **täckningsbidrag per timme** och Olles **tidrapport**.

Applikationen

Applikationen skiljer sig från övriga system genom att den innehåller programkod. Anvecklare använder dock sällan programmering för att utveckla kalkylsystem. Ett skäl till detta kan vara att formalisering genom programmering, upplevs som onaturlig av många anvecklare. En anvecklare som använder programmering för att tillgodose informationsbehov är fortfarande anvecklare så länge som han/hon tillgodoser egna informationsbehov i den egna verksamheten. Exempel på Applikationen är budgeteringssystemet som beskrivs i kapitel 6.

Indelningsgrunder

Den ovan beskrivna indelningen i kalkylsystemtyper utgår från anvecklarens verktygskunskaper. Denna indelning kan uppfattas som särskilt relevant då olika kunskapstyper relateras till anveckling. Andra indelningar som fokuserar andra orsaksförhållanden är förstås möjliga. Panko & Halverson Jr (1996:5) skisserar några alternativa indelningsgrunder:

- Användningsfrekvens (Engångssystem - Flergångssystem)
- Användarfrekvens (Enpersonssystem - Flerpersonssystem)
- Beslutstyp (Personliga beslut - organisationsbeslut)

Panko & Halverson Jr (1996:5) klassificerar även kalkylsystem utifrån "importance" och "materiality". Med *Importance* avses i vilken grad beslut bygger på information från kalkylsystemet och med *materiality* avses den ekonomiska betydelse informationen från kalkylsystemet har för företagets beslut. De indelningsgrunder som presenteras av Panko & Halverson Jr och av Floyd, Walls & Marr har relevans för kalkylsystem och andra typer av informationssystem. Indelningen utifrån anvecklares verktygskunskaper är däremot endast relevant för anveckling.

8.4 Human-Scale Information System

En modell som beskriver verksamhetsföreträdarens självständiga användning av datorer är Nurminens HIS-modell (Nurminen 1987). HIS uttyds Human-scale Information

System, vilket syftar till att representera ett humanistisk synsätt, som står i (viss motsats)relation till synsätt av systemteoretisk och socioteknisk natur. Informationssystem av HIS-typ överensstämmer väl med anvecklade kalkylsystem.

Idealtypen för det humanistiska synsättet kännetecknas av systemdesintegration, datorns avobjektifiering, människan i centrum och datorn som verktyg. I den rena HIS-världen finns inga delade databaser. Nurminen talar om *det personliga systemet*.

"Each worker has his or her own personal system, whose limits are defined in terms of that person's job. Such workers were formerly called 'users', presumably because of the integrated nature of the system they were expected to use." (Nurminen 1987:126)

Det personliga systemet behöver inte vara isolerat från omvärlden, det kan fungera som en del av en större helhet, men samordningen sker på andra grundvalar än vad som är fallet för det systemteoretiska och det sociotekniska systemet. Nurminen delar upp sin beskrivning av HIS i följande sex s.k. paradigmatiska begrepp:

- kunskap
- människa
- handling
- kommunikation
- organisation
- systemarbete (berörs ej i detta kapitel)

I och med sitt fokus på personligt användande av datorn finns en nära koppling till anveckling/KPA. Nedan diskuteras denna koppling i termer av nämnda paradigmatiska begrepp som Nurminen förknippar med HIS. Begreppet systemarbete berörs i kapitel 10. För diskussion av problem med HIS, se Avdic 1995a:169ff.

Kunskap

Kunskap finns i HIS endast i förhållande till subjektet människan. I och med att subjektet placeras i centrum blir syftet med informationen det centrala. Syftet blir det som uppfattas och förstås av subjektet. Informationen tolkas av subjektet, som också är mottagare av informationen. Informationen är direkt kopplad till sin användning. Informationen i datorn behöver, i HIS-system, inte nödvändigtvis vara heltäckande och konsistent. Systemets personliga natur tillåter individuella varianter av databehandling, överföring och lagring.

Kalkylsystem står ofta i relation till en människa och är ett uttryck för denna människas intentioner. Kalkylsystem kan också utvecklas i grupp av flera personer med gemensamma arbetsuppgifter och på så vis stå i relation till en speciell arbetsuppgift (Avdic1995a:180). I dessa fall smälter system och arbetsuppgift samman, vilket leder till att tolkningsproblem i stor utsträckning undviks eftersom informationen härstammar ur syftet med arbetsuppgiften. Kunskapsöverföringsproblemet finns inte i denna typ av system förrän i allra sista ledet. Slutliga mottagare (för definition av *mottagare*, se kap 11.3) av information är normalt andra personer än utvecklaren.

En relation mellan HIS-synsättet och KPA är att det är den person som är ansvarig för informationen som utformar kalkylsystemet, vilket inte gäller traditionell systemutveckling. Om systemet genererar felaktig information är det uppenbart att ansvaret fortfarande är anvecklarens. Det är fortfarande syftet med anvecklarens arbetsuppgifter som styr utformning av utdata. Den som känner detta syfte (anvecklaren) är dessutom den som utformar systemet. Anvecklarna i kapitel 3 och 4 är exempel på detta.

I ett idealt HIS-system hämtas inte information från andra system med automatik (se Kommunikation nedan). I en kalkylsystemmiljö, som har tillgång till ett lokalt nätverk kan dock centralt lagrad information utnyttjas. Även om anvecklaren via kalkylsystemet utnyttjar indata från fleranvändarsystem så kan det finnas möjligheter att identifiera informationens avsändare i första och andra led (Avdic 1995a:135). För anvecklaren är detta förhållande självklart och obligatoriskt. Att acceptera okända data som ej går att verifiera vore uteslutet för anvecklare. Anvecklarna i kapitel 3 och 4 strävar i själva verket efter noggrannhet. Det är ett av skälen till att de har blivit anvecklare.

Människa

Människan är i det humanistiska perspektivet utgångspunkten för informationssystem och systemutveckling. Arbetets mening härstammar från människan själv.

"If the job or the task is a meaningful one, neither sticks nor carrots are needed to induce the worker to perform it." (Nurminen 1987:131)

I HIS-perspektivet är det individen med sina arbetsuppgifter som är integrerad i stället för informationssystemet. Då Nurminen behandlar personliga system och människan så menar han människan och inte enbart människans arbetsuppgifter. Då det gäller kalkylsystem (Avdic 1995a:165) kan det inte sällan vara så att en människa och hennes arbetsuppgifter står i ett ett-till-ett förhållande till varann, vilket gör dessa kalkylsystem till HIS-system.

Handling

Den som handlar enligt det humanistiska synsättet är alltid människan. Även datorns tre grundläggande funktioner lagring, behandling och överföring av data kan ses som handlingar som en människa utför med sitt verktyg datorn. En (humanistisk) handling kräver mänsklig närvaro, en aktör. Ansvarsförhållanden är mindre oklara i det humanistiska perspektivet än i t.ex. det systemteoretiska perspektivet. Det är inte "datans fel" att en person presenterar felaktiga siffror, lika lite som det är hammarens fel att snickaren slår sig på tummen.

Handlingsaspekten gäller kalkylsystem i hög grad. Enligt HIS så kräver en handling att det finns en aktör. Anta att ett personligt kalkylsystem skulle ge felaktiga utdata. Om uppföljningen skulle bli missvisande, hur kan man anklaga datorn? Det är ju en person som skapat de beräkningar som leder till utdata. Det är ju till och med så att det är samma person som har skapat kalkylen som efterfrågar den informationen som produceras av systemet. Ansvarsförhållandet blir okomplicerat. Att det finns ett personligt ansvar för systembygget leder till ett speciellt förhållningssätt vid utvecklandet av systemet. Blir det fel får användaren själv ta ansvar för konsekvenserna. Om t.ex. Bosses

produktkalkyl (se kap 3.4) visar felaktiga värden kan ingen annan än Bosse hållas ansvarig för detta. Viss skillnad är det beroende på vem som tar initiativ till systemet, om personen är handlingsansvarig, formulerare eller införmedlare (se sid 215). Om den handlingsansvarige gör systemet till en införmedlare blir inte systemet renodlat integrerat med en person och en arbetsuppgift.

Kommunikation

All kommunikation sker mellan människor. Datorn ses inte som ett subjekt i informationssystemet. I de fall som system kommunicerar, (ett icke-HIS-uttryck) sker det alltid på så vis, att den mänskliga mottagaren av information har tillgång till den mänskliga avsändaren, som kan förklara och rätta till eventuella oklarheter. En uppdelning kan göras i inofficiell och officiell kommunikation.

"...it is unofficial communication which typically serves to maintain social reality." "This social reality in turn plays an important role in the creation of intersubjectivity and shared understanding, without which no cooperation or collaboration is possible." (Nurminen 1987:137)

Skillnaden mellan de två typerna av kommunikation blir mindre viktig i miljöer som präglas av det humanistiska perspektivet, därför att informationen i det individuella humanistiska systemet aldrig får den officiella prägel som ett integrerat systemteoretiskt präglat system får.

Ett kalkylsystem kan fungera så att verksamhetsföreträdaren registrerar indata och tolkar utdata. Kommunikation sker då från verksamhetsexperten till omgivningen. I de fall som en grupp utnyttjar systemet finns alltid människan närvarande och kan förklara och ställa till rätta, vilket kan vara en viss garanti för att datorn inte kommer att uppfattas som en kommunikationspartner. Då indata hämtas till kalkylsystemet från ett annat datorsystem, kan dock den verkliga kommunikationspartnern oidentifieras. Även om uppgifter kommer från ett annat datorsystem där andra personer matar in dessa uppgifter kan det finnas möjlighet att kommunicera "människa-till-människa". Om införmedlaren har verksamhetskunskap och har möjlighet att göra rimlighetsbedömningar, kommer han ändå inte att acceptera indata utan ha möjlighet att verifiera dessa indata (Avdic1995a:167). Då lokala nätverk används för distribution av information är anonymitet hos avsändaren knappast något som man eftersträvar. Detta berörs av Bosse i kapitel 3.4, som ser kalkylprogramutveckling som en kommunikationsfunktion. I de fall som relateras är det dock självklart att avsändaren (Bosse) finns tillgänglig för att förklara och rätta eventuella fel. I och med att kalkylsystemen finns tillgängliga digitalt kan mottagaren dessutom själv verifiera resultat och beräkningar.

Organisation

Eftersom humanistiska system följer arbetsorganisationen, följer informationsstrukturen den organisatoriska indelningen av en organisation, eller som Nurminen uttrycker det:

"Thus the idea of the organizational consequences of information technology can be seen as based on a fundamental misunderstanding of the situation." (Nurminen 1987:140)

Samordning sker precis som när det gäller kommunikation mellan människor. Om samordningen sker mellan namngivna personer blir den synlig och begriplig. Föreställningen att införandet av ett informationssystem alltid leder till organisationsförändringar gäller inte i den humanistiska världsbilden. Frågan om hur de personliga systemen skall länkas samman blir identiska med frågor om delning av arbete och samordning mellan individuella arbetsuppgifter. Datacentraler i systemteoretisk mening finns ej en renodlad HIS-miljö.

Kalkylsystem kan finnas i stand-alone miljö eller i nätverksmiljö. Arbetsplatser utan någon form av lokalt nätverk torde dock i Västeuropa bli allt ovanligare. Administration av lokala nätverk kräver (1999) någon form av administration. Utöver nätverksadministration kan man tänka sig ytterligare funktioner för en central administration, som kan underlätta anvecklarens verksamhet. Nedan följer några exempel på detta:

- Hjälp med hanteringen av maskinvara, t.ex. installation av skrivare
- Hjälp med hantering av programvara, t ex hur man skapar en matrisformel
- Hjälp med hantering av kommunikation mellan programvaror, som t.ex. hur man anropar program y från program x

Som rådgivande resurs, vad gäller datorer och programvaror, fyller alla personer som har kännedom om maskin- och programvara en uppgift. Dessa personer behöver inte vara anslutna till en organisatoriskt fristående datorcentral, lika lite som personer med t.ex. specifika språkkunskaper behöver vara anknutna till en språkcentral. Om stor- och minidatorer finns i organisationen förändras förutsättningarna. Stor- och minidatorer kräver specialistkompetens. Och i de fall som verksamhetsexperten av någon anledning behöver tillgång till data behövs specialisterna, åtminstone i ett inledningskede. Stor- och minidatorer är oförenliga med HIS-konceptet men är en realitet i många organisationer. I praktiken kommer möjligheterna att lösa verksamhetsorienterade problem i samband med utvecklande av personliga system, där man är beroende av stordatorkompetens, att bero på den policy som tillämpas på datorcentralen. Med detta menas att även om organisationen i ett företag är uppbyggd kring en datorcentral, kan personalen på denna datorcentral agera på ett sätt som stödjer en miljö där användare kan utveckla arbetsintegrerade, och därmed också HIS-orienterade (kalkyl)system.

HIS-system och kalkylsystem har stora likheter i och med att människan och hennes (integrerade) arbetsuppgifter är systemens utgångspunkter snarare än möjligheterna att integrera information i syfte att fördela och koordinera arbetsuppgifter i en organisation. En jämförelse mellan HIS och kalkylsystem visas i Tabell 8 nedan.

Tabell 8 Jämförelse av HIS och kalkylsystem.

Egenskaper	HIS	Kalkylsystem
Kunskap	Finns i förhållande till människan. Syftet avgörande	Finns i förhållande till en arbetsuppgift. Syftet avgörande
Människan	Människan är utgångspunkten	Människans arbetsuppgift är utgångspunkten
Handling	En handling kräver en aktör	En handling kräver en aktör
Kommunikation	Skär mellan människor	Skär oftast mellan människor. Undantag: indata från fleranvändarsystem
Organisation	Följer arbetsorganisationen	Följer arbetsorganisationen

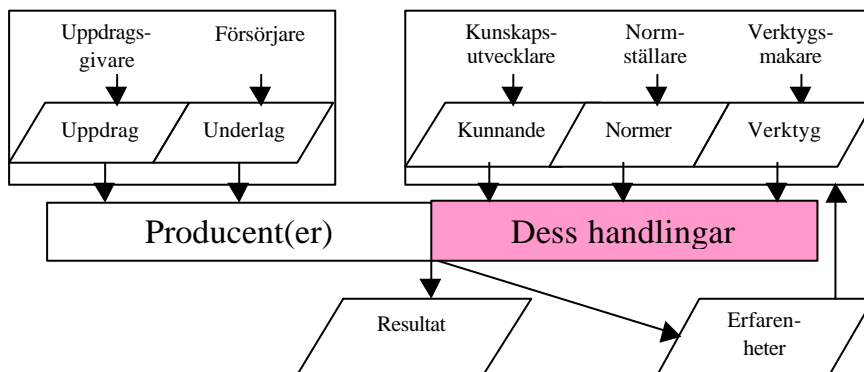
8.5 Sammanfattning informationssystem kalkylsystem

Kalkylsystem är en form av a-system som är en form av informationssystem. Ett kalkylsystem är ett informationssystem, som är utvecklat med hjälp av kalkylprogram och som fungerar integrerat i kalkylprogrammet. Utvecklade kalkylsystem kan relateras till Nurminens HIS-modell. Typiskt för kalkylsystem är att de är starkt relaterade till människor och dessas arbetsuppgifter. Systemen utmärks av integration av information och människors arbetsuppgifter. Skillnaden mellan TIS och kalkylsystem ligger mer i dess tillkomst och funktion än i själva utformningen. Ett system utvecklat av en IT-specialist till en användare med hjälp av kalkylprogram är sålunda ett TIS och inte ett kalkylsystem.

"Systemutvecklingsarbete... innebär människors arbete med att analysera, utforma och förändra verksamheter där datasystem ingår eller planeras ingå som delar."
(Goldkuhl 1993:25)

- *Detta kapitel syftar till att analysera systemutvecklingsbegreppet, i termer av traditionell systemutveckling och livscykelmodellen och relatera till kalkylprogramutveckling.*

9 Systemutveckling



Processen som leder till utformandet av ett informationssystem (IS) kallas här *systemutveckling*. Systemutveckling är ett vitt begrepp som kan uppfattas på olika sätt beroende på vilket perspektiv som intogs. Nedan diskuteras olika egenskaper som kan uppfattas hos systemutveckling och vilka som kan vara relevanta vid diskussion av traditionell systemutveckling (TSU) och anveckling.

9.1 Perspektiv på systemutveckling

Systemutveckling innebär *utveckling av datoriserade informationssystem*. Ett datoriserat informationssystem är en artefakt som automatiserar någon form av (del)rutin. Utvecklingen är de aktiviteter som leder fram till informationssystemet. Processen kan te sig på många olika sätt. Det kan vara fråga om allt från ett flerårigt projekt med hundratal aktörer inblandade till en aktivitet gjord på några minuter av en person.

Ett sätt att identifiera perspektiv på systemutveckling är att utgå från Andersens (1994:342) strategier för systemutveckling. Andersen identifierar sex egenskaper för systemutveckling och ytterlighetsvärden som dessa egenskaper kan anta (se Tabell 9). Nedan jämförs traditionell systemutveckling (TSU) och kalkylprogramutveckling över dessa egenskaper.

Tabell 9 Egenskaper för systemutveckling (Andersen 1994:342).

Egenskap	Ytterlighet 1		Ytterlighet 2
Egeninsats	Systemutveckling	↔	Standardsystem
Typ av metod	Analytisk	↔	Experimentell ¹
Leverans	Revolutionär	↔	Evolutionär
Användarmedverkan	Expertdominerad	↔	Användarorienterad
Typ av resultat	Produkt	↔	Process
Samordning	Ensidig systemutveckling	↔	Pso-utveckl. ²

Andersens indelning gäller främst TSU men innehåller ändå ett antal egenskaper kan användas för att fördjupa förståelsen för anveckling. Den första egenskapen, *egeninsats*, är dock inte lika tillämplig för anveckling som övriga egenskaper. Å ena sidan kan man säga att KPA är systemutveckling såtillvida att utvecklaren konstruerar ett, om än litet, system. Samtidigt är kalkylprogrammet självt en helt avgörande del i kalkylsystem, vilket gör att kalkylsystem skulle kunna betraktas som en variant av standardsystem. Det producerade kalkylsystemet kan i många stycken uppfattas som ett modifierat kalkylprogram, vilket innebär att det är ett slags standardiserbart, programmerbart standardsystem. Kalkylprogram intar en mellanställning vad gäller utvecklingsmiljöer, se t.ex. Figur 90, där tre teknologinivåer identifieras vid utveckling av beslutsstödsystem. Även här kan kalkylprogram klassificeras som positionerat mellan å ena sidan färdiga applikationer och å andra sidan verktyg för utveckling av applikationer. Den mellanställning som antyds ovan är en viktig orsak till att kalkylprogram används vid anveckling. Ett traditionellt utvecklingsverktyg, t.ex. C++, har för hög inlärningströskel och ett standardsystem är inte tillräckligt flexibelt för anveckling av den typ som exemplifierats i kapitel 3, 4 och 6. Arbetssätt vid KPA beskrivs i Avdic (1995a:180ff) och sammanfattande på sid 189.

När det gäller de övriga kategorierna kan KPA överlag placeras in i den senare av värdeparen. För *typ av metod* gäller att KPA är utpräglat experimentell. Utvecklare begagnar sig inte av systemutvecklingsmetoder för att utveckla kalkylsystem, vilket ofta är fallet vid TSU. Typiskt för KPA är istället stegvis förfining där systemet utvecklas efterhand som arbetsuppgifterna erfordrar. För egenskapen *leverans* rör det sig för KPA normalt om evolutionära systemutvecklingsprocesser, medan implementeringen ofta sker revolutionärt vid TSU. Idén med en leverans av systemet till väntande beställare och användare är vid KPA definitionsmissigt otänkbar. För *användarmedverkan* gäller att KPA definitionsmissigt är fullständigt användarorienterad, eljest vore

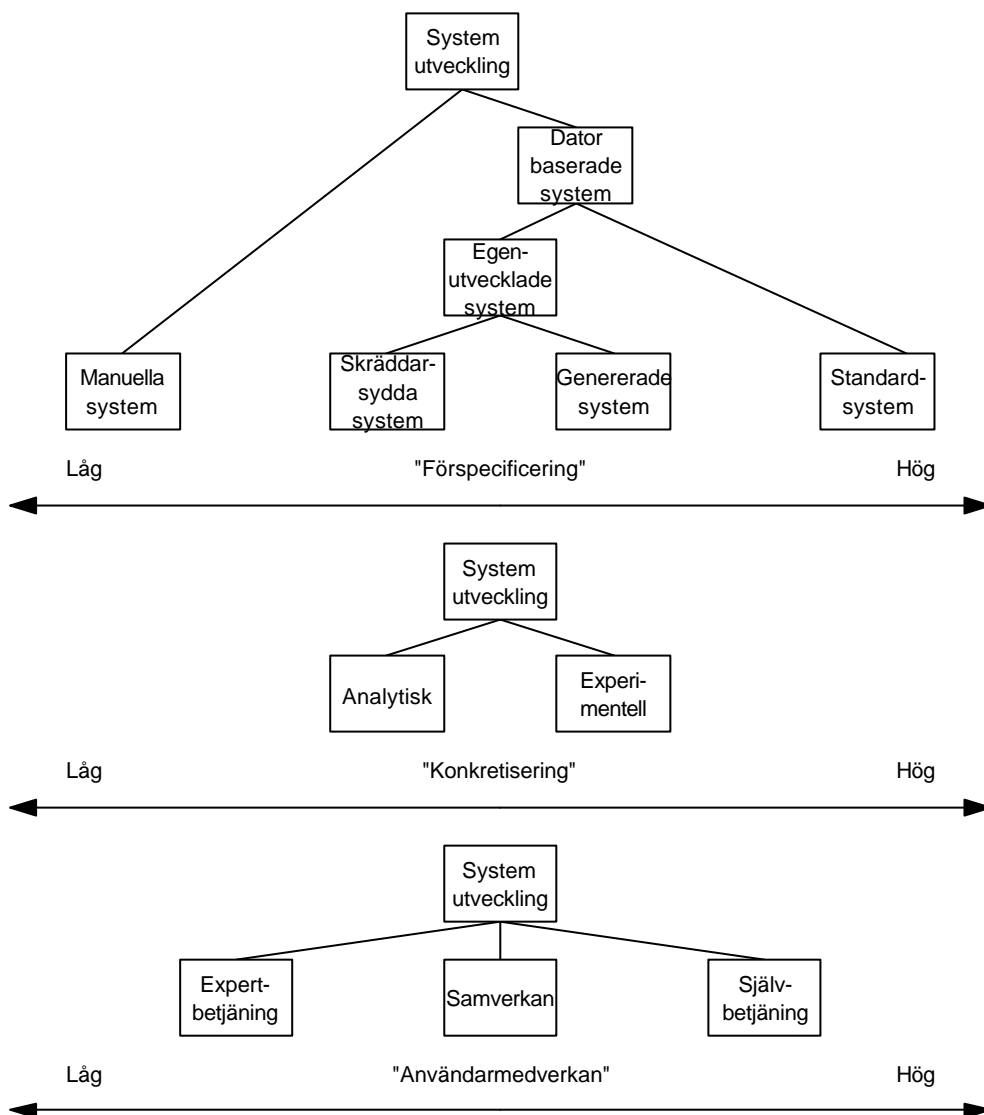
¹ Denna kategorisering benämner Nilsson (1991:57) grad av konkretisering.

² Personnel (Staff), information-System, and Organization.

det inte fråga om anveckling. För *typ av resultat* gäller att processen ofta är väsentlig vid KPA eftersom det ofta rör sig om en lärandeprocess, utdragen i tiden. Att utesluta att produkten i sig kan vara det väsentliga är dock inte möjligt. Ismos **beräkning av arkantal per pall** kan ses som ett fall av produktorienterad KPA, då det tillverkades snabbt för att användas i färdigt skick. Slutligen kan KPA, när det gäller *samordning*, karaktäriseras som pso-utveckling i och med att det är en arbetsintegrerad process, ej skild från verksamheten i övrigt. Om förhållandet vore det motsatta när det gäller de fem sista kategorierna (analytisk, revolutionär, expertdominerad, produktorienterad och ensidig systemutveckling) kan det hävdas att det är en form av "traditionell" systemutveckling. Vad som utmärker "traditionell" systemutveckling diskuteras närmare nedan. Av de egenskaper, som nämns ovan, kan sägas att samtliga är meningsfulla att tillämpa på TSU medan KPA inte kan vara vare sig analytisk, revolutionär, expertorienterad eller ensidigt systemutvecklande.

Nilsson (1991:57) klassificerar systemutveckling genom att klassificera utvecklingsprocessen i kategorierna *grad av förspecifisering*, *grad av konkretisering*, och *grad av användarmedverkan* (se Figur 80 nedan). Nilssons klassificering innehåller inte lika många kategorier som Andersens modell, men är något mer finkornig i vissa avseenden. När det gäller *grad av förkategorisering*, delar Nilsson upp egenutvecklade system i skraddarsydda och genererade system. På samma sätt som när det gäller Andersens första kategori så blir KPA/anveckling inte självklar att passa in. Kalkylsystem är på sätt och vis genererade i och med att en del av funktionaliteten finns inbyggd i kalkylsystemet. På samma sätt som anförs ovan kan kalkylprogram sägas vara till viss del skraddarsydda och till viss del standardsystem. Svårigheten att passa in KPA i denna klassificering beror till stor del på den integrerade karaktär som kännetecknar KPA. Kalkylsystem har låg förspecifisering vad gäller systemutvecklingsarbetet men utnyttjar samtidigt starkt förspecifiserade funktioner i kalkylprogrammet. Ett exempel på när kalkylprogrammets standardfunktioner använts i hög utsträckning är Olles **pris- och indexutveckling** (se Figur 52 sid 96) som består av en tabell och ett diagram som visualiserar utvecklingen av för verksamheten centrala index. Här har kalkylprogrammets standardfunktioner utnyttjats i hög grad. Exemplet i Figur 48, **tidrapport**, visar ett system där utvecklingsarbete (utan förspecifisering) tagit större utrymme. Bland respondenterna i såväl D1 som D2 finns en stor variation i kalkylsystemens grad av förspecifisering. Vad gäller *konkretisering* är KPA, som nämnts ovan definitionsmässigt experimentell. Anvecklingens arbetsintegrerade karaktär innebär att utvecklingsarbetet styrs av linjearbetsuppgifterna, vilket står i viss motsats till det projektorienterade utvecklingsarbete som kännetecknar TSU. Å andra sidan finns det inget som hindrar anvecklaren att initiera ett förändringsprojekt som går ut på att skapa ett kalkylsystem som stöd för den egna verksamheten. I de studerade fallen är applikationen i kapitel 6 det enda exemplet på ett sådant förfarande. När det gäller *användarmedverkan* är KPA, närmast synonymt med självbetjäning. Samtidigt kan en typ av samverkan äga rum vid KPA/anveckling när det gäller stöd för kommunikation via nätverk, stöd för installation av kringutrustning och hjälpprogram. Den självständiga systemutvecklingsform som anveckling/KPA innebär skapar nya behov av samverkan med IT-specialister. Någon expertbetjäning är det dock aldrig fråga om i anvecklingssamma n-hang.

Användare och utvecklare



Figur 80 Olika typer av systemutveckling (Nilsson 1991:57).

En sammanfattande kategorisering av KPA och TSU återfinns i slutet av detta kapitel.

9.2 Traditionell systemutveckling (TSU)

För att relatera KPA till befintlig teori, beskrivs här s.k. traditionell systemutveckling (TSU). Uttrycket ger inte helt rättvisa åt det fenomen som skall framställas som jämförelseobjekt. Uttrycket "traditionell" är inte avsett att tolkas bokstavligt. Användningen av uttrycket TSU syftar till att precisera en vanligt förekommande syn på systemutveckling. Preciseringsen är inte avsedd att kritisera någon form av systemutveckling, inte heller finns det någon ambition att värdera ett arbetssätt som bättre än ett annat. Då uttrycket systemutveckling är vanligt förekommande i litteratur inom området, finns det skäl att identifiera någon form av idealtyp, representerande en specifik form av systemutveckling som kan användas som jämförelsemått.

Begreppet systemutveckling är centralt på samma sätt som begreppet informationssystem och är i likhet med många andra begrepp inom ADB-området inte entydigt defi-

Systemutveckling

nierat. Olika människor lägger olika betydelse i uttrycket systemutveckling. För en del har det en teknisk anklang medan det för andra har en mer organisatorisk karaktär. Ingen kan förneka att det är förknippat med praktisk utveckling av datorbaserade informationssystem, men författare och forskare understryker i allmänhet vikten av analys och utredning. I de allmänna modeller som ligger till grund för systemutvecklingsmetoder finns en inledande analysfas med.

En definition som beskriver vilka *arbetsuppgifter* som ingår i systemutvecklingsarbetet ges av Andersen (1994:42):

"Man inleder systemutvecklingsarbetet med att planera informationssystemet. Planeringsarbetet kallas systemering." "Systemutvecklingen består av systemering, realisering och implementering."

Andersen hävdar att en konsekvens av distinktionen mellan systemering och systemutveckling är att de ger orsak till skilda arbetsuppgifter. Denna syn står i rak motsats till KPA, som utmärks av integration av arbetsuppgifter i utvecklingsarbetet. Anvecklaren inte bara planerar, utan konstruerar, förvaltar och använder även systemet. Ett exempel på detta är Jans **bokslutssystem** (se sid 104).

Organisatoriska och verksamhetsorienterade aspekter kan betonas i samband med utveckling av system, såväl traditionella som anvecklade. Bansler (1990) skriver t.ex.

"Begreppet systemutveckling är för det första knutet till situationer där utveckling och införande av *datasystem* spelar en väsentlig roll - även om det inte framgår av själva ordet. För det andra är det som regel underförstått att målet inte bara är att utveckla datasystem i snäv mening, utan att förändra de befintliga arbetsprocesserna och den befintliga arbetsorganisationen inom en viss verksamhet eller institution. Systemutveckling kan därför betraktas som en form av *organisationsförändring* snarare än en teknisk konstruktionsprocess." (Bansler 1990:5f)

En annan definition på systemutveckling, som förutom angivande av arbetsuppgifter även ytterligare betonar den verksamhetsorienterade aspekten med systemutvecklingsarbete, lyder:

"Människors arbete med att analysera, utforma och förändra verksamheter där datasystem ingår eller planeras ingå som delar." (Goldkuhl 1993:25)

Den senare definitionen betonar särskilt verksamhetens roll. Informationssystem utvecklas inte för sin egen skull, utan är en typ av verksamhetsutveckling, som syftar till att en verksamhet bättre kan uppfylla sina mål med hjälp av systemet. Att bedriva systemutveckling är att verksamhetsutveckla datasystem (Goldkuhl 1993:25). Den senare definitionen utesluter inte utveckling av kalkylsystem som systemutvecklingsverksamhet. I själva verket är denna typ av verksamhet verksamhetsutvecklande eftersom det är verksamhetens krav som driver anvecklaren att utveckla kalkylsystem för att lösa problem i sin verksamhet. På samma sätt som den handlingsorienterade definitionen av begreppet informationssystem skiljer sig från den traditionella, skiljer sig den senare definitionen av systemutveckling från den traditionella på så vis att den betonar systemutveckling som en integrerad del av en verksamhet och dess uppgifter.

Definitionerna ovan antyder att systemutveckling är en aktivitet som försiggår bland många människor under en icke försumbar tid. Systemutveckling beskrivs traditionellt som en aktivitet som försiggår i projektform (Andersen 1994:111, Sundgren 1992:41, Yourdon 1989:77ff, m.fl.). Det uppfattas även som underförstått att det är något som några gör åt några andra. Detta kan bero på ett medvetet ställningstagande att vissa informationssystem, t.ex. kalkylsystem inte är att betrakta som informationssystem. Det kan också bero på att definitionerna och beskrivningarna av systemutveckling ursprungligen härrör från en tid då tekniken att bygga informationssystem var så komplicerad att det endast var förbehållet ett fåtal utvecklingsexperter. Min utgångspunkt är den senare. Man har helt enkelt inte i definitionerna införlivat tanken på att nya miljöer tillåter icke-IT-specialister att bygga professionella applikationer, som helt eller delvis ersätter de mer traditionella informationssystemen. Om det skulle vara ordningsföljden på aktiviteterna eller vilka personer som genomför systemutveckling som avgjorde vilka aktiviteter som skulle få kallas systemutveckling, då skulle de aktiviteter som användare utför för att utveckla kalkylsystem inte kunna kallas systemutveckling, även om man utför motsvarande aktiviteter och utvecklar en produkt med samma funktionalitet.

Traditionell systemutveckling

I avhandlingen används uttrycket *traditionell systemutveckling* som en referenspunkt till *anveckling/KPA*. Med traditionell systemutveckling avses systemutveckling som utförs av IT-specialister för verksamhetsföreträdare. Följande egenskaper kännetecknar traditionell systemutveckling:

- Utveckling sker under en avgränsad tidsperiod.
- Konstruktion sker efter kravspecifikation utformad i en analysfas.
- Utvecklingen av systemet utförs av utvecklingsspecialister
- Då systemet är klart sker ett överlämnande
- Realiseringen sker normalt med hjälp av programutvecklingsverktyg som kräver specialkunskaper, t.ex. programspråk som Cobol, C++, Delphi eller Visual Basic
- Förvaltning av systemet utförs av speciell utvecklingspersonal
- Utvecklingsmetoder tillämpas som anpassade till stora, komplexa fleranvändarsystem.

9.3 Livscykelmodellen

Livscykelmodellen eller *Vattenfallsmodellen* är vanliga metaforer då TSU diskuteras eller beskrivs. I denna avhandling framställs och används Livscykelmodellen som en idealtyp för TSU-arbete. Livscykelmodellen beskriver systemutveckling som en sekvens av faser, vilka påbörjas i en viss ordning. Modellen förekommer i olika sammanhang i olika skepnader. Det finns även varianter av systemutvecklingsmodeller som gör smärre avsteg från idealtypen. De flesta modeller har en fasindelning av aktiviteter, som kan sägas utgöra specialfall av Livscykelmodellen. Livscykelmodellen representerar en IT-specialistorienterad syn på SU-arbete. För att användare skall kunna delta meningsfullt i livscykelorienterade processer måste utbildning av användarna äga rum. Livscykelmodellen har starka kopplingar till vad som här benämns traditionell systemutveckling, dvs systemutveckling som bedrivs i projektform under en av-

Systemutveckling

gränsad tidsperiod av projektgrupper och av IT-specialister. System utvecklade efter Livscykelmodellen realiserar normalt med hjälp av programspråk (av typ Cobol, C, Visual Basic, C++ e.d.), 4GL-verktyg (t.ex. Paradox, dBase, Access eller varför inte kalkylprogram) eller med standardsystem (Nilsson 1991). Att kalkylprogram kan klassas som fjärdegenerationsverktyg innebär inte att all systemutveckling med hjälp av dessa sker i enlighet med Livscykelmodellen.

En av de första indelningarna i metodområden i systemutvecklingsarbetet utarbetades av Börje Langefors (1967). Livscykelmodellen har stora likheter med Langefors indelning:

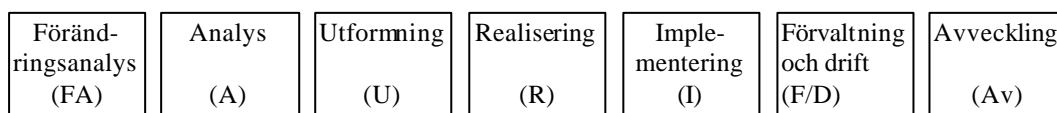
1. Objektsystemanalys (Verksamhetsanalys)
2. Informationsanalys
3. Datasystemutformning (Datasystemkonstruktion)
4. Realisering, implementering och drift (Langefors 1967:86)

Langefors indelning innehåller en infologisk (1 och 2) och en datalogisk (3 och 4) del. De infologiska områdena berörde informationsaspekter (Vad) medan de datalogiska områdena handlade om konstruktion och realisering (Hur).

En väsentlig del av den litteratur som finns om systemutveckling utgår från en livscykelsyn (Berg & Hultman 1985, Axelsson & Ortman 1985, Cameron 1989, Yourdon 1989, Andersen 1994, Nilsson 1991, Sundgren 1992, Goldkuhl 1993, Beekman 1997 etc.). Livscykelmodellen kan ses som en slags måttstock för analys av systemutvecklingsmetoder och -aktiviteter. I själva verket innehåller alla traditionella beskrivningar av systemutveckling någon form av livscykelmodell. Typiskt för dessa modeller är att de ser systemutveckling som en följd av faser som skall påbörjas i tur och ordning. Iterationer och parallellt arbete kan mycket väl förekomma, men den första fasen påbörjas före den andra, den andra påbörjas före den tredje osv.

Beskrivningen ovan av Livscykelmodellen skall ses som en principiell beskrivning avsedd att fånga huvuddragen i en vanligt förekommande tankemodell som representant för en vanlig syn på systemutvecklingsarbete. Avsteg från Livscykeln kan förekomma i olika sammanhang t.ex. visar Nilsson (1991:65) att vid införande av standardsystem kan i vissa situationer faser utelämnas. Under senare år har prototyping blivit vanligt förekommande i systemutvecklingsprocessen (Andersen 1994:405ff, Budde m.fl. 1992). Detta kan också i viss utsträckning ses som ett avsteg från livscykelmodellens olika faser.

Den variant av Livscykelmodellen som används i denna avhandling som utgångspunkt och jämförelseobjekt och som redovisas i figuren nedan är hämtad från Andersen (1994:41). Andersens livscykelmodell består av de sju faserna i Figur 81 nedan.



Figur 81 Livscykelmodellen (Andersen 1994:41).

Beskrivning och resultatet av Livscykelmodellens faser kan sammanfattas enligt Tabell 10 nedan. (De sista två faserna kan inte sägas generera något specifikt resultat).

Användare och utvecklare

Tabell 10 Beskrivning av faserna i livscykelmodellen och dess resultat enligt Andersen (1994:43).

Fas	Beskrivning	Resultat
FA	Utredning av verksamhetens problem och möjligheter Arbetsuppgifter består i att beskriva nuläge, önskad situation, behov och åtgärder Deltagare: Verksamhetsledning, verksamhetsmedarbetare och konsulter	Valda utvecklingsåtgärder
A	Analysarbetet delas upp i <i>Verksamhetsanalys</i> (VA) och <i>Informationssystemanalys</i> (IA) VA: Analys av verksamhet samt avgörande hur IS kan underlätta verksamhet Deltagare: Användares chefer, användarrepresentanter och systemerare IA: Bedömning och bestämning av IS-innehåll Deltagare: Användare och systemerare	Kravspecifikation
U	Utformning delas upp <i>Principiell utformning av teknisk lösning</i> (PUT) och <i>Utformning av utrustningsanpassad teknisk lösning</i> (UT) PUT: Bedöma och bestämma principiell teknisk lösning Deltagare: Systemerare och programmerare UT: Välja utrustning, bedöma och bestämma praktisk lösning Deltagare: Systemerare och programmerare	Realiserbart informationssystem
R	Utarbetande av informationssystemet Utarbeta dataprogram och manuella rutiner Deltagare: programmerare	Färdigt informationssystem
I	Start av systemet Ta nya program och rutiner i bruk Deltagare: Användare, systemerare och programmerare	Infört Informationssystem
F/D	Underhåll och förbättring av systemet. Göra korrigeringar. Bedöma behov av och eventuellt genomföra förbättringar. Drift.	
Av	Avveckling av systemet, säkring av information	

Andersen grupperar faserna på så vis att han menar att Analys och Utformning tillsammans utgör *Systemering*. *Systemutveckling* menar Andersen är faserna Analys, Utformning, Realisering och Implementering. Denna uppdelning innebär att Förändringsanalys inte betraktas som en del av systemutveckling samtidigt som den (Förändringsanalysen) förutsätts föregå utvecklingsarbetet. Enligt Andersen är Livscykelmodellen det mest framgångsrika tillvägagångssättet "...vid många utvecklingsuppgifter".

Andersen (1991:32ff) och andra (t.ex. Flensburg 1986) använder husbyggnation som liknelse för att beskriva systemutveckling. Liknelsen går ut på att då man bygger hus måste man först ta ställning till vad huset skall användas till (Förändringsanalys), sedan utforma en grov plan över vad huset t ex skall ha för sorts rum (Analys), därefter

Systemutveckling

vidtar ritningsarbete (Utformning), byggnation (Realisering) och idrifttagande (Implementering). Efter byggnationen är huset klart att användas (Förvaltning och Drift). Utveckling skulle kunna motsvaras av rivning. Liknelsen har pedagogiska förtjänster men då man driver den för hårt bortser man från vissa grundläggande skillnader mellan husbyggnation och systemutveckling t.ex. möjligheten till prototyping och tillgången till slutanvändarverktyg. Vad gäller nämnda faktorer så kan det vara svårt att hitta en motsvarighet till prototyping vid husbyggnation. Slut användarverktyg av typ kalkylprogram skulle närmast motsvaras av ett färdigbyggt hus med mycket stora möjligheter till anpassningar, vilka skulle kunna utföras av den som skall bo i huset. Ovan beskrivna modell bygger på följande grundantaganden (ungefär som husbyggnation).

- människan är en förnuftig varelse som kan betrakta sin omgivning och planera rationellt
- systemutveckling utförs av personer med specialkunskaper
- man planerar i detalj innan man konstruerar
- system utvecklas från grunden
- utveckling av informationssystem är en komplicerad och tidsödande aktivitet
- utveckling sker utifrån de förutsättningar som stordatormiljö skapar.

Det kan sägas vara uttryck för rationalistiska vetenskapsideal som leder till:

- aktivitetssuppdelning
- rolluppdelning
- specialisttänkande
- sekvenstänkande
- totalutveckling
- storskalig teknik.

Att husbygge jämförs med systemutveckling är ett uttryck för en syn på systemutveckling som en projektstyrd aktivitet där det är viktigt att göra saker i rätt ordning och där det är svårt att rätta till fel som görs i ett inledningsskede, vilket gör de inledande faserna principiellt viktiga. T.ex. skulle det enligt husbyggnadsliknelsen vara omöjligt att börja ett husbygge med att tapetsera, vilket det ju också är. När det gäller informationssystemkonstruktion "kan man börja med tapeterna", vilket är en effekt av informationens väsen. Då denna effekt utnyttjas systematiskt kan det kallas för Prototyping.

Beskrivningar är centrala i Livscykelmodellen. Utformningen av beskrivningarna följer den Top-down-ansats som Livscykelmodellen i sig själv följer. Enligt Andersen växer det blivande informationssystemet fram "...genom arbete med olika slags beskrivningar, där man först behandlar överordnande frågor och till slut övergår till beskrivningar av detaljerade tekniska förhållanden" (Andersen 1994:40). En annan omständighet som Andersen framhåller som typisk för Livscykelmodellen är deltagande av olika slags experter. Andersen skiljer på systemutveckling och systemering och därmed också på dessa roller i systemutvecklingsprocessen.

9.4 Livscykelmodellen - kalkylprogramanveckling

Jämförelsen mellan Livscykelmodellen och KPA ter sig intressant med tanke på att Livscykelmodellen framstår som en idealtyp för TSU. En signifikant skillnad mellan TSU och KPA är hur utvecklingsarbetet bedrivs. Medan systemutveckling enligt Livscykelmodellen sker huvudsakligen sekventiellt så har inte systemutveckling enligt KPA någon självklar startpunkt. KPA framstår överhuvudtaget inte som en specifik process med en början, en mitt och ett slut, även om det i och för sig kan vara så. KPA utmärks av att utvecklingsarbetet är integrerat i arbetsuppgifterna för en person eller en grupp personer. Detta leder dels till att utvecklingsarbetet kan pågå parallellt med linjearbetsuppgifterna och inte behöver vare sig, realiseras, implementeras eller avvecklas som specifika, från arbetsuppgifterna isolerade moment. En KPA-aktivitet kan ta sin början i en realisering av en mindre del av det framtida kalkylsystemet. I Avdic (1995a:151) ges ett exempel på detta. Exemplet handlar om en deltagarlista som så småningom utvecklades den till ett **resultatberäkningssystem**. Man skulle även kunna säga att Förändringsanalys till viss del kan bedrivas integrerat i KPA. Förståelsen för verksamhetens problem och möjligheter kan utvecklas parallellt med utvecklandet av kalkylsystemet. Ofta kan kalkylsystemet ändras allteftersom problemuppfattningarna förändras.³ Diskussionen leder till slutsatsen att KPA kan sägas vara ett instrument för (kontinuerlig) förändringsanalys. Ett exempel på detta från Avdic (1995a:151) gäller kalkylsystemet **budgetuppföljning** kan sägas vara ett exempel på hur problemuppfattningarna kan förändras. Då utfall jämförs med budget i diagramform blir såväl resultatbegreppet som budgetbegreppet mer påtagligt än då jämförelsen sker med siffror enskilda dagar. Den visuella uppdelningen på dagar och i jämförelse med budget och tidigare utfall ledde till fördjupad analys av de största avvikelserna. Presentation i diagramform uppfattades som åskådlig och tillgänglig för alla aktörgrupperna i produktionen, som var målgrupp för utdata. I kapitel 3 finns flera exempel på när Förändringsanalys integreras i hela utvecklingsprocessen, t.ex. Bosses **täckningsbidrag per timme**. Det som skrivs ovan utgår från en uppfattning att förändringsanalys alltid föregår förändring men att den kan vara explicit i olika grad. Den förändringsanalys som beskrivs i Livscykelmodellen är organiserad och dokumenterad medan den som bedrivs i KPA ofta är oorganiserad och odokumenterad. Överhuvudtaget framstår dokumentation och beskrivningar som mindre framträdande i KPA än i livscykelinfluerade systemutvecklingsprocesser.

Livscykelmodellen som sådan är relevant vid diskussion kring KPA, då den pekar ut generella moment som genomförs vid systemutveckling. Alla faser i Livscykelmodellen kan sägas genomföras då KPA bedrivs, fast eventuellt i en annan ordning (se Tabell 11 nedan).

³ Men inte alltid! Då kalkylsystemet blir för komplext kan det givetvis vara svårt att ändra.

Systemutveckling

Tabell 11 KPA relaterat till livscykelmodellen (Avdic 1995a:152f).

Fas	Motsvarighet i KPA
FA	<p>Kan utföras separat innan KPA eller integrerat i linjearbetsuppgifterna eller parallellt/integrerat i det övriga utvecklingsarbetet.</p> <p>Förändringsanalys sker dock normalt som en naturlig del av anvecklarnas ordinarie linjearbete. Enligt Livscykelmodellen är detta en aktivitet som specialister gör för verksamhetsrepresentanter. Ett av skälen till att det görs är att specialis-terna måste beskriva detta för att förstå det överhuvudtaget. Denna process (att förstå) är inte nödvändig då anvecklaren utvecklar för egna behov.</p>
A	<p>Detta moment sker i KPA integrerat i utvecklingsarbetet.</p> <p>Det utförs inte på traditionellt sätt med grafer och andra beskrivningar. Analys under KPA bedrivs integrerat i utvecklingsarbetet, eftersom anvecklaren är specialist framför allt på sin verksamhet ligger avgörandena kring lämplighet inbakade i hans ordinarie linjearbete. Dokumentation av analysresultat fyller här inte samma funktion som enligt livscykelmodellen där den delvis är till för kommunikation mellan användare och utvecklare. Exempel på detta är vilken sort en viss indatacell skall innehålla.</p>
U	<p>Detta moment sker i KPA till viss del integrerat i utvecklingsarbetet.</p> <p>Bedömning av principiell teknisk lösning innebär att värdera vilka möjligheter som programvarumiljön tillåter vad avser t.ex. datastrukturer, inmatning, bearbetning och utdata. I en KPA-aktivitet sker här ett växelspel mellan anvecklarens önskemål och kunnande om verktyget. T.ex. kan en insikt om verktygets möjligheter föda en idé om en utveckling av stöd för vissa arbetsuppgifter.</p> <p>När det gäller val av utrustning så är ett PC-baserat kalkylprogram alltid utgångspunkten. Det kan dock visa sig att uppgiften inte går att lösa inom dessa ramar. Skälet till detta kan vara låg prestanda, bristande kompetens etc.</p>
R	<p>Detta moment sker i KPA integrerat i utvecklingsarbetet.</p> <p>Realisering är ett teknikbetonat moment i livscykelmodellen, som för användare kan framstå som en svårbegriplig aktivitet. I KPA bedrivs detta arbete genom att skapa formler, funktioner och diagram av olika slag. Kompetensen som behövs för att utforma enkla kalkylsystem kan förvärfvas på någon dag. För mer komplexa system krävs mer kunskap.</p> <p>Att jämföra realisering enligt Livscykelmodellen och enligt KPA ger en av de mest slående skillnaderna mellan dessa två synsätt, eftersom detta moment är ett av de moment i Livscykelmodellen som är mest expertorienterat.</p>
I	<p>Detta moment sker integrerat i utvecklingsarbetet.</p> <p>Startar systemet gör man inte vid någon särskild tidpunkt i KPA. Momenten utförs i den ordning som anvecklaren finner det lämpligt, vilket kan betyda att kalkylsystemet kan tas i drift helt eller delvis redan efter inskrivning av en formel eller en text.</p>

F/D	<p>Även detta moment sker i KPA integrerat med utvecklingsarbetet. Att tala om drift och förvaltning underförstått innebär att ett system kan betraktas som färdigt och därefter utsättas för förändringar om nya krav dyker upp i omvärlden. Detta synsätt är mer begripligt i en värld där system byggs projektorienterat och där olika roller figurerar i utvecklingsarbetet.</p> <p>I KPA sköter anvecklaren såväl drift som förvaltning inom ramen för sitt ordinarie linjearbete och det uppfattas inte av honom som en systemutvecklingsaktivitet utan snarare som ett utförande av ordinarie arbetsuppgifter.</p> <p>De delar av driften som har att göra med backup på gemensamma sekundärminnen och byte av programvara, dvs sådant som inte går att knyta naturligt till ordinarie arbetsuppgifter kan mycket väl utföras av en dataavdelning precis som när det gäller Livscykelmodellen.</p>
Av	<p>Om Avveckling sker arbetsintegrerat så innebär det att kalkylsystem rensas bort och att information säkras om så är nödvändigt. Detta kan utföras av anvecklaren eller av dataavdelningen.</p>

Systemutformningshjälpmedel innehåller moduler och funktioner som i olika utsträckning är färdigkonstruerade för den person som har att utforma system. Frekvensen och omfattningen av dessa moduler och funktioner varierar med hjälpmedlet. En Cobol-kompilator innehåller kanske funktioner för databasanrop, medan man i mer maskinnära miljö tvingas utveckla dessa eller andra funktioner inom ramen för utformningsfasen. När det gäller utformning med kalkylprogram så är en mängd funktioner (t.ex. diagram- och rapportfunktionerna) i stort sett färdiga att utnyttja för anvecklaren, vilket innebär att utformningsarbetet till viss del redan är utfört av programvaruleverantören och endast till viss del utförs av anvecklaren. Nämnade förhållande påverkar speciellt utformnings-, realiserings- och förvaltning/driftsfaserna i relation till Livscykelmodellen.

Livscykelmodellen är tillämpbar på traditionell systemutveckling som bedrivs i projektform med tredje generationens verktyg av typen Cobol, Pascal eller C++. Verktygen kräver specialistkompetens för att kunna utnyttjas effektivt. Modellen är också tillämpbar på utveckling med fjärde generationsverktyg samt på upphandlingsmodellen för standardssystem.

KPA sker i PC-miljö med hjälp av kalkylprogram. Kalkylprogrammet är till sin karaktär ett slutanvändarverktyg, vilket innebär att man inte behöver vara IT-specialist för att kunna använda det, vilket i sin tur är en förutsättning för KPA. Utveckling enligt Livscykelmodellen förknippas knappast med slutanvändarverktyg (Andersen 1994:48f). Visserligen kan kalkylprogram användas som utvecklingsverktyg i mer traditionell mening, vilket också görs, men då är det inte fråga om KPA. Utvecklingen av kalkylsystem av typen *Applikationen* kan ha inslag av traditionell utveckling (enligt Livscykelmodellen), men också inslag av KPA då användaren med kunskap om verktyget deltar aktivt i alla delar av utvecklingsarbetet och även hanterar och förvaltar systemet aktivt. Exempel på detta kan vara utformning och förändring av mallar för indata och utdata samt mindre förändringar i programkod. Kapitel 6 ger ett exempel på denna form av användarprototyping.

9.5 Sammanfattning Livscykelmodellen - kalkylprogramanveckling

Livscykelmodellen är inte relevant för KPA som en modell över den *process* som utförs, däremot kan de moment som Livscykelmodellen pekar ut användas för att analysera KPA som fenomen.

KPA innebär en syn på systemutveckling, som skiljer sig från den syn som Livscykelmodellen representerar på flera plan. Flera av modellens begrepp blir irrelevanta då de skall tillämpas på KPA. Särskilt gäller detta eftersom Livscykelmodellen representerar en syn på systemutveckling som föddes då maskin- och programvara skapade andra förutsättningar för systemutvecklingsarbete.

Kriteriet för vilka systemutvecklingsuppgifter som är relevanta för KPA är huruvida uppgifterna är förenliga med den ordinarie arbetsuppgiften. Kopplingen till ordinarie arbetsuppgifter gör att systemutvecklingen som aktivitet är underordnad de ordinarie arbetsuppgifterna. KPA representerar i relation till Livscykelmodellen en utveckling mot integration av utvecklings- och arbetsuppgifter. I Tabell 12 nedan sammanfattas jämförelsen mellan Livscykelmodellen och KPA.

Tabell 12 Jämförelse Livscykelmodellen - KPA, efter Avdic (1995a:156)

Jämförelsepunkter	Livscykelmodellen	KPA
Systemutveckling i relation till bedrivande av verksamhet	Separation. Utvecklingsarbete sker separat från löpande linjearbete.	Integration. Utvecklingsarbete sker integrerat med löpande linje arbete.
Arbetsfördelning olika faser i systemutvecklingsprocessen	Specialisering. Livscykels faser utförs av olika personer med olika kompetens.	Integration. Samma person utför olika faser i livscykeln.
Förändringsanalys	Separation. FA utförs separerat från övriga utvecklingsarbetet.	Integration. FA sker normalt integrerat i utvecklingsarbetet men <i>kan</i> ske separat.
Systemtyporientering	Komplexitet. Anpassad för komplexa fleranvändarsystem.	Enkelhet. Anpassat för företrädesvis enkla system och enanvändarsystem.
Processstyp	Sekvens. Faserna påbörjas i turordning.	Integration. Stegvis förfining Ingen specifik ordning mellan olika moment.
Jämförelsepunkter	Livscykelmodellen	KPA
Processens varaktighet	Projektorientering.	Arbetsintegration.

Användare och utvecklare

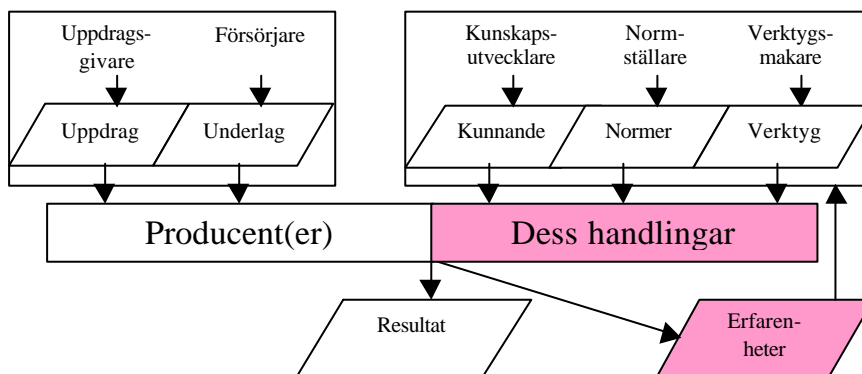
Rollfördelning aktörer (utvecklare-användare)	Separation. Utvecklare och användare har skilda roller.	Integration. Utvecklar- och användarroller integreras.
Utvecklare	Systemutvecklings-specialist.	Anvecklare.
Leverans	Leverans efter utveckling.	Användning och utveckling integrerat.
Miljö	Egenutveckling eller standardssystem. Fleranvändarmiljö.	PC-miljö, kalkylprogram. En- eller fleranvändarmiljö.
Utvecklingsverktyg	Verktyg som kräver kodkunskap.	Direktmanipulerande eller deklarativa verktyg.
Metoder	Avpassade för stora, komplexa fleranvändarsystem.	Inga standardiserade metoder.

Anveckling

"The implicit assumption seems to be that end-users are untrained programmers who, given the opportunity, will enthusiastically deluge the organization in 'spaghetti' code, running on incompatible machines, in the same way that professionals did until recently. Thus, it seems to follow that they should be controlled and trained. Whilst there may be a few isolated instances of this type of behavior ... we would regard this as a rather one-sided and in some ways dated view."
(Smithson & Hirschheim 1990:38)

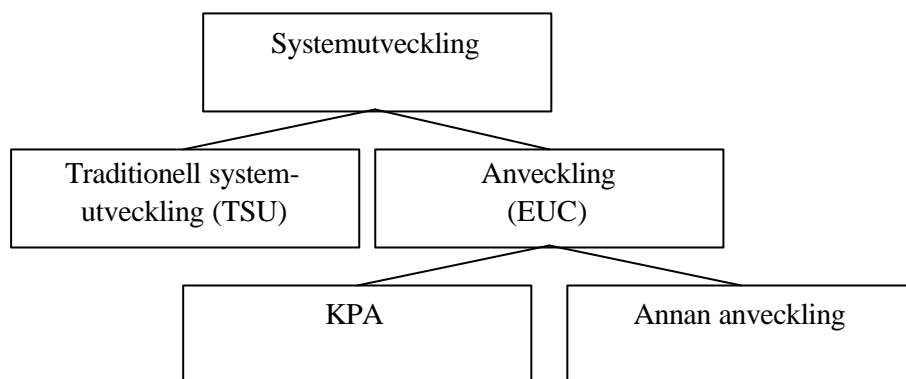
- *Detta kapitel syftar till att analysera anvecklingsbegreppet, samt relatera det till relevant forskning.*

10 Anveckling



I kapitel 9 behandlas systemutvecklingsprocessen ur ett perspektiv som kan appliceras på olika typer av systemutveckling, speciellt på traditionell systemutveckling (TSU), men även (kalkylprogram)anveckling. Eftersom kalkylprogramanveckling (KPA) är en speciell form av systemutveckling (se Figur 82 nedan) med egna egenskaper i förhållande till TSU, tarvar denna form av systemutveckling ett eget kapitel. Fenomenet KPA, kan närmast relateras till forskningsområdet *End User Computing* (EUC) eftersom det behandlar verksamhetsföreträdarens användning av IT (se definition sid 187). EUC är ett brett område, som i viss mån skiljer sig från forskningsområdet informationssystemutveckling. Forskning om systemutveckling sker traditionellt utifrån ett systemutvecklingsperspektiv, som fokuserar på systemutveckling som fenomen, där datorer och system ofta är själva föremålet för forskningen. Eftersom forskning inom EUC-området ofta sysslar med användning av datorer utifrån ett "datorer som medel"-perspektiv har kunskapsutvecklingen om EUC en annan karaktär än den som gäller

TSU. Då referenser om EUC söks handlar dessa ofta om styrning av datoranvändning ur ett ledningsperspektiv.



Figur 82 KPA är en typ av anveckling som är en typ av systemutveckling.

Begreppen *EUC* och *anveckling* är synonyma. Båda skulle kunna användas, dels som benämningar på problemområden och dels som benämningar på aktiviteter inom detta problemområde. I avhandlingen används överlag uttrycket *anveckling* av skäl som presenteras i kapitel 1. Uttrycket *EUC* används dock emellanåt i detta kapitel då det förekommer bland referenserna till kapitlet.

Detta kapitel representerar enligt den praktikgeneriska modellen en processaspekt på *anveckling*. I kapitlet behandlas dock fler aspekter av *anveckling* än själva processen. Dessa aspekter skulle ha kunna fördelats på de olika kapitlen i referensramsdelen (*Del III*) av avhandlingen. Istället kommer forskning om *anveckling*, som bedöms vara relevant för *anveckling/KPA*, att presenteras samlat i detta kapitel. Två undantag från detta är områdena *aktörer* och *verktyg*, som behandlas under respektive kapitel. Redovisningen av *anvecklingsrelaterad* forskning kommer i detta kapitel grovt att följa den praktikgeneriska modellen genom att innehållet i kapitlet delas in i grupperna *Förutsättningar-Praktik/process-Effekter*. Först diskuteras och definieras begreppen *anveckling* (kap 10.1) och *KPA* (kap 10.2), därefter presenteras några perspektiv på *anveckling* och *förutsättningar* för *anveckling* (kap 10.3). Efter detta följer en diskussion om effekter och problem med *anveckling* (kap 10.4). I praktikdelen (kap 10.5) återfinns en diskussion om metodik för *anveckling*. Slutligen görs en sammanfattning i kapitel 10.6. Det kunnande och de verktyg (och eventuellt normer), som *anvecklaren* utvecklar i sin praktik återgår, i form av *erfarenheter*, till *anvecklaren* som *förutsättningar* för nästa *anvecklingsaktivitet*. Kunnande, normer och verktyg behandlas i kapitel 13, 14 och 15.

10.1 Anveckling: definitioner och terminologi

Anveckling handlar om verksamhetsföreträdarens användning av datorer för eget bruk. *anveckling* är en av flera benämningar på ett fenomen som definieras på olika sätt av olika personer.¹ Definitionerna skiljer sig bl.a. åt genom att olika mycket betona huru-

¹ Den vanligaste är nog End User Computing (EUC).

Anveckling

vida användare verkligen utvecklar system.² Enligt den vidaste definitionen skulle även självständig användning av datorer i största allmänhet kunna räknas som anveckling. Ett exempel på detta är följande definition:

"...autonomous use of information technology by knowledge workers outside of the information systems department." (Storey & Chan 1994:1).

och

"...the use of computer hardware and software by people in organizations whose jobs are usually classified as net users of information systems rather than net developers of information systems." Conger (1992:274)

En definition som mer betonar applikationsutvecklingsaspekten är följande:

"...the adoption and use of information technology by personnel outside the information systems department to DEVELOP software applications in support of organizational tasks." (Brancheau & Brown 1993:437)

Andra definitioner med liknande innebörd finns hos Parker, Verner & Cerpa (1993:353) och Carlsson (1993:1). Den senare definitionen (Brancheau & Brown) används som utgångspunkt i denna avhandling, eftersom ren användning av datorer, som åsyftas i de första definitionerna ovan, blir en definition som innefattar i det närmaste all typ av användning.

Innebörden av anveckling formuleras av Lally (1995) som "applications developed by novices for their own use and applications developed by professional analysts and programmers for others."³ Hon framhåller följande kännetecken för "End User Computing":

1. Novices can do their thinking before or after they do their programming. No pre-planning or specifications which must be approved by the user are required.
2. Novices know what is expected of the system.
3. Novices do not need to have changes authorized.
4. Novices are interested in gaining new insight into their problem, not in becoming a better programmer.
5. Novices do not need to create documentation.

(Lally 1995:4)

Programmering enligt Lally behöver inte nödvändigtvis betyda kodprogrammering. Det kan lika gärna röra sig om deklarativ regelformulering. Lallys karakterisering poängterar en pragmatisk aspekt, anvecklare bestämmer över sin arbetssituation, vet vad han vill ha, fokuserar på problemet och utelämnar sådant som inte upplevs som nödvändigt för lösandet av problemet. Detta kan upplevas som positivt eller negativt. För somliga i verksamheten är detta önskvärt och uppfattas leda till en kreativ verksamhetsutvecklande aktivitet med hjälp av ny teknik, medan det för andra framstår som en vildvuxen aktivitet som bör styras och kontrolleras för att inte skada verksamheten. På samma sätt behandlas anveckling av forskare och författare. anveckling, som det definieras ovan, är ett fenomen som tycks ha kommit för att stanna, men synen på hur det skall bedömas och hanteras varierar.

² I bemärkelsen utveckling av informationssystem (om än små sådana ibland).

³ I Lallys indelning kallas anvecklaren för novisen, vilket kan tolkas som ett uttryck för ett specialistperspektiv. Specialisten på utveckling är ju novis på verksamheten, vilket dock inte framhålls.

Anveckling och EUC som benämning är inte ensamt förekommande på det aktuella fenomenet. Andra benämningar med liknande betydelse är

- Desktop Information Technology (Kaiser & Oppelland 1990)
- Desktop Computing (Davies & Wood-Harper 1990)
- Personal Computing (Gogan 1990, Isakovitz m.fl. 1995)

Svenska benämningsvarianter är:

- Personlig databehandling (Flensburg 1986)
- Självbetjäning (Nilsson 1991:56ff)
- Användarledd systemutveckling (Andersen 1994:350ff).

Det faktum att begreppsfloran är varierad antyder att området inte omfattas av standarder och ändamålsenliga begrepp. En annan tänkbar förklaring är att utvecklingen inom området går så snabbt att betydelsen av benämningar inte hinner stabiliseras innan förutsättningarna har förändrats. När det gäller utvecklingen av kalkylprogram är detta relativt påtagligt. På elva år (1988-1999) har karaktären på kalkylprogram förändrats väsentligt, vilket har lett till att även förutsättningarna för KPA förändrats. Formuleringarna av ytterligare ett uttryck (anveckling) och motiven till detta återfinns i kapitel 1.

Enligt Nurminens (1988:143) HIS-modell utmärks systemarbete enligt HIS (se sid 165) av förenkling, delvis som en reaktion på traditionella systemutvecklingsmetoder, vilka ur ett HIS-perspektiv är otillräckliga på flera sätt. Begreppet användarmedverkan blir irrelevant då man talar om systemutveckling med humanistiskt synsätt. HIS-system är ju personliga och en integrerad del av individens personliga arbetsuppgifter. I och med att arbetet och dess mening har en så framträdande roll garanteras "användaren" en central roll i utvecklingsarbetet. System som utvecklas, likaväl som utvecklingsprojekt, är av begränsad storlek och komplexitet. Förändringar kan eventuellt utföras av "användaren" själv. Utvecklingen kan ske stegvis och under en längre tidsperiod i takt med att "användaren" lär sig tekniken. Denna syn på systemarbete har många paralleller med KPA (se nedan).

Begreppen *anveckling* och *kalkylprogramanveckling* är definierade på sid 24.

10.2 Kalkylprogramanveckling

KPA är ur anveklarens perspektiv huvudsakligen att betrakta som utförande av arbetsuppgifter i en verksamhet, snarare än systemutveckling. Å andra sidan är KPA en form av systemutveckling i och med att systemutvecklingsaktiviteter utförs och i och med att produkten är ett informationssystem (se Figur 78 sid 162). I detta avsnitt sammanfattas vad som utmärker KPA enligt Avdic (1995a).

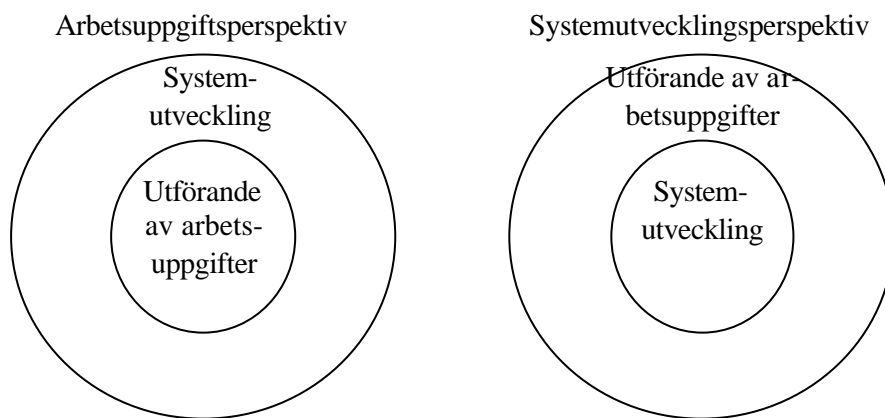
Systemutveckling ur ett arbetsuppgiftsperspektiv

Kalkylprogramanveckling (KPA) kännetecknas av att systemutveckling utförs av någon (anveklaren) utifrån ett *arbetsuppgiftsperspektiv*. Ett arbetsuppgiftsperspektiv innebär fokuserande på specifika arbetsuppgifter och dessas mål. Ur ett arbetsuppgiftsperspektiv är det arbetsuppgiften som är central och inte systemutvecklingsaktiviteten som sådan. Verktyg med vilka arbetsuppgiften utförs kan i princip vara vilka som helst så länge som den bidrar till att uppfylla syftet med arbetsuppgiften. Att fo-

Anveckling

kusera på arbetsuppgiften innebär att endast aspekter i utvecklingsarbetet som har att göra med arbetsuppgiften behandlas. Det är t.ex. inte säkert att dokumentation och indatakontroller intar en lika fokuserad position som det förutsätts göra vid TSU. Skälet till detta är anvecklarenas kunskaper om verksamheten och dess uttalade och outtalade mål.

Arbetsuppgiftsperspektivet står i motsats till systemutvecklingsperspektivet, (se Figur 83 nedan) som ur systemutvecklarens synvinkel utmärker arbetet vid TSU. Systemutvecklingsperspektivet innebär att systemutvecklingsarbetet fokuseras i förhållande till de arbetsuppgifter som är föremål för systemutveckling. Uppgiften som systemutvecklingsarbetet gäller, kan i princip vara vilken som helst. Det som skiljer den professionella systemutvecklaren från anvecklaren är just förhållandet till verksamhetens arbetsuppgifter som primärt eller sekundärt.



Figur 83 Olika fokus på arbetsuppgifter och systemutveckling beroende på perspektiv (Avdic 1995a:180).

KPA är följaktligen huvudsakligen utförande av arbetsuppgift samtidigt som det är fråga om systemutveckling. Denna omständighet gör att KPA kan klassificeras som *arbetsintegrerad systemutveckling*. Att fokusera systemutveckling ur ett arbetsuppgiftsperspektiv är ett mål med denna avhandling.

Arbetssätt

KPA utmärks av:

- Stegvis förfining
- Sammanflätade aktiviteter
- Icke-procedurell utveckling
- Avsaknad av standardiserat metodarbete (Avdic 1995a:180ff)

Med *stegvis förfining* menas att arbetet bedrivs iterativt och inkrementellt, vilket innebär att kalkylsystemet konstrueras efterhand som behov upplevs och att utformningen sker aktiviteter utförs om och om igen tills systemet uppfattas som ändamålsenligt. Med *sammanflätade aktiviteter* sammanflätade aktiviteter menas att utveckling, användande och lärande sker parallellt. Utveckling och användning sker av kalkylsystemet. Lärande berör såväl verktygskunskap som verksamhet. Fem av de sex respondenterna i kapitel 3 och 4 framhåller just kunskapsutveckling om verksamheten som

ett viktigt mål med anveckling. Att lärande om verktyget var viktigt framgick vid intervjuerna då kalkylsystemens funktionalitet diskuterades flitigt. De tre aktiviteterna är för anvecklaren integrerade och sinsemellan beroende. Att anvecklaren arbetar med *icke-procedurell utveckling* innebär att kalkylsystemet normalt ej utvecklas med traditionella utvecklingsverktyg utan snarare med användande av deklarativa eller direktmanipulerande verktygsfunktioner. Alla exempel i kapitel 3 och 4 är exempel på detta. Undantag är exemplet i kapitel 6. Med *avsaknad av standardiserat metodarbete* menas att den metodtradition som är central inom TSU inte kommer till användning vid KPA. Den arbetsintegrerade aspekten väver samman utvecklingsarbete och arbetsuppgifter på ett sätt som inte underlättar användande av projekt- och dokumentationsorienterade ansatser. En del av syftet med systemutvecklingsmetoder, att verifiera verksamhetens uppb yggnad, kan även förefalla mindre nödvändig för anvecklaren.

Den kalkylprogramorienterade aspekten

Kalkylprogram präglas av kalkylbladets centrala roll och dess möjlighet till beräkningar och grafisk presentation. Ett alternativ till kalkylprogram som anvecklingsverktyg är databashanteringsprogram⁴. Under 1990-talet har funktionaliteten hos dessa två utvecklingsverktyg alltmer kommit att överlappas. Greve och Karlsson (1994:205) menar att anvecklare har lättare att utnyttja kalkylprogram. En undersökning av Holmberg (1996) visar att kalkylprogram och databashanteringsprogram i mångt och mycket har liknande funktionalitet. Det som utmärker kalkylprogram är den omedelbara tillgängligheten för den ovane användaren samt något mer lättillgänglig möjlighet till beräkningar och till grafisk presentation.

Tillgänglighetsaspekter

Tillgänglighet till kalkylprogram är en viktig förutsättning för KPA (Avdic 1995a:202). Liknande slutsatser drar Brancheau & Brown (1993:460). Då man jämför tillgänglighetsförhållandena på 1980-talet med motsvarande på 1990-talet tenderar denna förutsättning att uppfyllas allt oftare, kalkylprogram blir alltså allt vanligare. Skälet till detta är spridningen av s.k. programpaket⁵. Till detta kan, då man jämför med databashanteringsprogram, även en ekonomisk aspekt läggas. Det är ännu så länge (september 1999) dyrare med databashanteringsprogram. I större organisationer kan skillnaden mellan att göra kalkylprogram och databashanteringsprogram tillgängliga för alla handläggare handla om icke försumbara belopp.

Innebörd av kalkylprogramanveckling: Decentralisering genom integration

KPA är ett uttryck för decentralisering av systemutvecklingsverksamheten i en organisation.(Avdic 1995a:199) Detta görs möjligt genom olika typer av *integration*. KPA innebär integration av:

- Verksamhetsansvar, verksamhetskunskap och utvecklingskompetens
- Systemutvecklingsaktiviteter
- Aktörsfunktioner
- Behandlingsfunktioner

⁴ T.ex. MS Access och Borland Paradox.

⁵ T.ex. Lotus SmartSuite och MS Office.

Anveckling

- Informationssystemets komponenter i kalkylsystemet

Verksamhetsansvar, verksamhetskunskap och utvecklingskompetens integreras på så vis att den som har informationsbehovet och ansvar för aktuell (del)verksamhet är den som utvecklar systemet.

Systemutvecklingsaktiviteter som integreras är (med Livscykelmodellen som referens) (Andersen 1994:39ff) förändringsanalys, verksamhets- och informationsanalys, utformning, realisering, implementering, förvaltning/drift och avveckling.

Aktörsfunktioner. Anvecklaren antar flera roller, IT-specialist, användare och beslutsfattare (se även kap 11). Bland IT-specialistroller kan utvecklingsroller skiljas från drift och förvaltningsroller. Med detta sägs inte att anvecklaren alltid sköter allt, t.ex. kan kunskap om nätverk och skrivare inhämtas från dataavdelning eller motsvarande.

Behandlingsfunktioner integreras i kalkylprogrammet, som innehåller funktioner för olika typer av behandling som anvecklaren kan tänkas behöva, t.ex. grafisk presentation, rapportering, beräkning, textbehandling och programmering.

Informationssystemets komponenter har av Langefors (1966) definierats som inmatning, bearbetning, lagring, överföring och presentation. I ett enkelt kalkylsystem kan samtliga dessa komponenter samlas på ett kalkylblad.

Integrationen, som innebär att systemutvecklingskompetensen decentraliseras, får ett antal följder. En konsekvens är att det s.k. kunskapsöverföringsproblemet minskar. En annan konsekvens är att ledtider för viss systemutveckling minskar. En tredje är att möjligheterna att tillgodose delar av verksamhetens informationsbehov ökar (se även kap 10.4). Integrationsaspekterna står i kontrast till TSU, som kännetecknas av att IT-specialister utvecklar system och att användare använder dem. Integrationen ger ny innebörd till flera centrala begrepp inom systemutvecklingsområdet, som t.ex. användarvänlighet, användardeltagande och kvalitet (Avdic 1995a).

10.3 Förutsättningar för anveckling

I detta avsnitt diskuteras förutsättningar och framgångsfaktorer för anveckling. Först behandlas hur spridning av anveckling kan gå till, därefter klassificeras förutsättningar och framgångsfaktorer enligt kategorierna möjlighet, berättigande och lämplighet.

10.3.1 Spridning av anveckling

I Avdic (1995a:119ff) visas att användning av kalkylprogram varierar i olika organisationer på flera sätt. Användning kan t.ex. variera avseende volym och komplexitet. Olika organisationer förefaller ha olika förutsättning för spridande av anveckling. I *End-User Computing: A Study of Three Management Models* redogör Parker, Verner & Cerpa (1993) för modeller för betraktande av hur anveckling sprids i en organisation. Enligt modellen *Four Strategies for EUC Growth* klassificeras det sätt som anveckling i en

organisation växer fram på, enligt faktorerna expansion och kontroll. Faktorn *kontroll* kan uppfattas som något som är beroende av mål i verksamheten och som kan styras. Faktorn *expansion* är däremot ett uttryck för omständigheter inom det område (geografiskt eller branschmässigt) en organisation verkar. Kontrollfaktorn är följaktligen utsatt för strategi och taktik och därmed ett uttryck för viljeyttringar. Huruvida en organisation strävar efter hög kontroll eller inte vad gäller anveckling påverkar vilka effekter som anses önskvärda eller ej (Parker, Verner & Cerpa 1993:355). Enligt *Applied Contingency Theory* är *kontrollstrategi* och förekomst av *professionalism* de viktigaste faktorerna för spridning av EUC. En strikt kontrollerad organisation utan större förekomst av professionella skulle kunna erbjuda mindre förutsättningar för anveckling än en icke-hierarkisk organisation med betydande förekomst av självständiga professionella. De utvecklare som bidragit med empiri till såväl D1 som D2, har haft kvalificerade arbetsuppgifter och stort ansvar för den egna verksamhet. Följaktligen kan den redovisade empirin styrka att anveckling kan bedrivas där professionalism finns och där kontrollen är låg.

Brancheau & Browns (1993)⁶ sammanställning av forskning om anveckling visar bl.a. att:

- Olika organisationer och olika utvecklare har olika lätt att anamma anveckling
- Mellanmänniska kanaler är viktigare än mer officiella kanaler vad gäller att sprida anveckling
- Uppfattade egenskaper hos anvecklingsverktyget viktigt (Se även kap 15)

Enligt studien visar empiriska och teoretiska data att det behövs mer än strategi och teknik för att göra styrning av anveckling effektiv. Enbart formella styrningsaktiviteter tycks inte ensamma vara tillräckligt effektiva. Speciellt gäller detta officiella kontrollprocedurer. Det finns, enligt Brancheau & Brown, övertygande belegg för att *informella nätverk* starkt påverkar beslut om anammande av utvecklingsverktyg, utvecklingskompetens, verktygsanvändning och PC-säkerhetsbeteende. Av de undersökta ledningsaktiviteterna tycktes *utbildning* vara den mest effektiva. Särskilt effektiv blev utbildningen när den tidsmässigt inträffade i kombination med efterfrågan på kunskaper och när den kombinerades med uppföljningsstöd samt när den uppmuntrades av normer inom arbetsgruppen (Brancheau & Brown 1993:463). I de studerade fallen har anveckling initierats och bedrivits av utvecklarna själva, t.ex. Ismo, Bosse och Thomas på Industri och Olle, Jan och Annette på Myndighet. Tanken att personer med självständiga arbeten med kontrollfunktioner skulle kunna styras att bedriva systemutveckling med kalkylprogram förefaller orealistisk.

Enligt ovan relaterad empiri och teori framstår följande faktorer som väsentliga för spridning av anveckling

- Kontrollpolicy
- Förekomst av professionalism
- Förekomst av informella kommunikationskanaler
- Utbildning
- Verktyg

⁶ Brancheau & Browns studie av forskning om anveckling har ett särskilt intresse eftersom den sammanfattar inte mindre än 90 teoretiska och empiriska studier av anveckling mellan 1983 och 1990. De studerade studierna har varit publicerade i ansedda tidskrifter huvudsakligen från Nordamerika.

10.3.2 Möjligt, berättigat, lämpligt

I detta avsnitt klassificeras förutsättningar och framgångsfaktorer för EUC efter kategorierna möjlighet, berättigande och lämplighet (Waterman 1985:127) (se *Bilaga 1*). Med *möjlighet* menas faktorer som krävs för att anveckling överhuvudtaget skall vara möjlig. Möjlighetsfaktorerna är av OCH-karaktär, vilket innebär att samtliga möjlighetsfaktorer krävs för att möjliggöra anveckling. Med *berättigande* menas faktorer som i någon mening kan berättiga anveckling. Berättigandefaktorerna är av ELLER-karaktär, vilket innebär att minst en berättigandefaktor krävs för att berättiga KPA. Med *lämplighet* menas faktorer som i något väsentligt avseende gör det direkt lämpligt att anveckla med kalkylprogram. Lämplighetsfaktorerna är av klustervis OCH-karaktär, vilket innebär att ett kluster av lämplighetsfaktorer krävs för att KPA skall vara lämpligt.

Nedan ges några exempel på hur de olika faktorerna fungerar utifrån KPA. Om en möjlighetsfaktor saknas är KPA omöjlig. T.ex. kan ett kalkylsystem vara omöjligt att bygga om det inte finns tillgång till ett kalkylprogram. Ett system kan vara möjligt men inte berättigat eller lämpligt då inte ett enda rationellt skäl talar för att det byggs med kalkylprogram. Ett exempel skulle kunna vara ett bokningssystem i realtid med många användare, som skulle kunna vara möjligt att konstruera med kalkylprogram utan att det för den skull vara vare sig berättigat eller lämpligt med tanke på t.ex. ekonomiska skäl och prestandaskäl. Ett system skulle kunna vara möjligt och berättigat men inte nödvändigtvis lämpligt om det finns minst ett rationellt skäl som talar för att det byggs med kalkylprogram. Ett exempel skulle kunna vara att anvecklaren har goda kunskaper i verktyget och att det därför skulle gå fortare att bygga systemet med kalkylprogram än med något annat utvecklingsverktyg även om andra verktyg skulle kunna uppfattas som lämpligare då man bortser från anvecklarens förkunskaper. Ett system är möjligt, berättigat och lämpligt att bygga med kalkylprogram om alla väsentliga skäl talar för att kalkylprogram används som utvecklingsverktyg. Vilka som är de väsentliga faktorerna för att göra KPA lämpligt kan olika forskare ha olika uppfattning om. Detta samt hur uppfattningarna om faktorerna skiljer sig diskuteras i detta avsnitt.

10.3.3 Möjlighet

Vilka faktorer som möjliggjort anveckling i allmänhet och KPA i synnerhet diskuteras i kapitel 1, här diskuteras möjlighetsfaktorer utifrån uppfattningar uttryckta av EUC-forskare.

Som grundläggande förutsättningar för anveckling anger Robey & Zmud (1990:15):

- en utbildad och datamogen grupp slutanvändare
- användarvänlig programvara
- fortsatt IT-utveckling

Även Andersen (1994:443) nämner nya verktyg som en faktor som har gjort anveckling möjlig.

I sin doktorsavhandling *A Longitudinal Study of User Developed Decision Support Systems* behandlar Sven A Carlsson (1993) användning och icke-användning av kalkylprogram för beslutstöd vad gäller design, användning och påverkan av användarbyggda beslutsstödssystem. Systemen är byggda med hjälp av kalkylprogram. Studien genomfördes med fjorton frivilliga, välmotiverade anställda i en offentlig verksamhet som med ledningens stöd gick en kurs i kalkylprogramanvändning. Av dessa fjorton kom endast tre att bli reguljära kalkylprogramanvändare. Carlsson menar utifrån studien att tre grundläggande förutsättningar för anveckling är:

- tillgång till ett lättanvänt datorverktyg
- utbildning
- tilltro till individens positiva attityd.

Utöver dessa förutsättningar formulerar Carlsson ytterligare tre som är nödvändiga för kalkylprogramanvändning:

- Verktyget⁷ och uppgiften som skall stödjas skall matcha varann.
- Uppgiften som skall stödjas skall vara väsentlig⁸.
- Den presumtive användaren skall ha självständighet, handlingsfrihet och makt att förändra uppgiften, dess ändamål samt dess aktiviteter.

Möjlighetsfaktorer

Möjlighetsfaktorer skulle utifrån ovanstående resonemang kunna vara

- *Maskinvara* som gör det möjligt att använda anvecklingsverktyg, t.ex. PC
- *Utvecklingsverktyg* med egenskaper som gör det möjligt, för en verksamhetsföreträdare utan specialkunskaper, att anveckla, t.ex. kalkylprogram
- Grundläggande *kunskaper* hos anvecklaren som gör det möjligt för denne att kunna anveckla, t.ex. grundläggande dator- och verktygskunskap
- En för anvecklaren *självständig arbetsituation* för att kunna anveckla, t.ex. respondenterna i kapitel 3 och 4
- *Uppgifter* som är möjliga för en anvecklare att ta sig an

Av de faktorer som inte tagits med betraktas inte "fortsatt IT-utveckling" och "tilltro till individens positiva attityd" som tillräckligt kritiska för att kunna klassificeras som möjlig- faktorer

10.3.4 Berättigande

Faktorer som ensamma eller tillsammans med andra kan berättiga anveckling innebär inte automatiskt att det då blir lämpligt.

Robey & Zmud (1990:15) pekar ut långa ledtider vid TSU och användarnas otålighet som orsaker till att anveckling tillgrips av användare. Detta resonemang innebär att TSU som sådant berättigar anveckling när TSU uppfattas som långsamt. I de två verksamheter som respondenterna i kapitel 3 & 4 bedrivs TSU såväl av den egna dataavdelningen som på konsultbasis. Alternativet att engagera extern kompetens för utveckling av de redovisade kalkylsystemen förefaller orealistiskt då dataavdelningarna varit

⁷ "the DSS generator".

⁸ "central and critical".

Anveckling

upptagna med projekt av mer omfattande dignitet. För t.ex. Jan kan anvecklingsverksamheten, t.ex. **bokslutssystem**, ses som ett sätt att utföra det ordinarie arbetet.

Anveckling är enligt Reynolds (1992:348) berättigat i *dynamiska verksamheter*⁹ eller verksamheter med *varierade informationsbehov*¹⁰. För fem av de sex respondenterna (Annette är möjligen undantagen) fanns inget uttalat krav på vilken information som förväntades att anvecklaren skulle prestera. De mål som styr anvecklingen har legat på ett plan som inneburit att olika sätt att utföra arbetet funnits att välja på. På myndigheten har kravet på nollresultat, kommunal demokrati och resurshushållning styr Olle och Jan i sin anvecklingsverksamhet. På industrin har kundnytta och resurshushållning styr Ismo, Bosse och Thomas.

Andersen (1994) kallar anveckling för användarledd systemutveckling (se kap 10.5). Andersen menar att användarledd systemutveckling inte får vara den enda typen av systemutveckling i en verksamhet. Den bör vara ett komplement till den traditionella systemutvecklingen där ADB-expertter deltar. Argument som talar för (berättigar) användarledd systemutveckling kan enligt Andersen vara:

- ADB-avdelningen hinner inte med
- Anvecklade system motsvarar bättre användares önskemål (Andersen 1994:439)

Berättigandefaktorer

Berättigandefaktorerna skulle utifrån ovanstående resonemang kunna vara faktorer som gör anveckling bättre lämpad än andra sätt att tillgodose informationsbehov för att utföra arbetsuppgifter. Orsakerna kan ha att göra med

- hur dynamisk verksamheten är (*verksamhetens föränderlighet*)
- hur föränderliga informationsbehoven är (*informationsbehovens föränderlighet*)
- hur lång tid det tar att producera efterfrågad information (krav på *informationens aktualitet*)
- hur väl den producerade informationen motsvarar beställarens önskemål (krav på *informationens kvalitet*)

Berättigandefaktorerna är orienterade mot verksamhetens karaktär och behov av information. Den mest grundläggande berättigandefaktorn är verksamhetens föränderlighet. I en dynamisk verksamhet som verkar på en föränderlig arena kan systemutveckling, som kan anpassas till denna föränderlighet vara berättigad. Eftersom anveckling är arbetsuppgiftsorienterad och följer verksamheten, eller snarare är en del av den löpande verksamheten kan det finnas skäl att tro att kraven på föränderlighet kan tillgodoses av anveckling.

Att anvecklingsverksamheten i de studerade verksamheterna framstår som såväl möjliga som lämpliga är inte så överraskande, då de valts med tanke på att ge möjlighet till studium av anveckling.

⁹ "The program logic, calculations, or report formats will be changed frequently".

¹⁰ "...need for ad-hoc reports".

10.3.5 Lämplighet

När förutsättningarna är sådana att anveckling är möjlig och berättigad är det ändå inte säkert att det är lämpligt. Nedan ges exempel på forskning där denna typ av förutsättningar diskuterats.

Rivard & Huff (1988) har formulerat en modell för faktorer för anveckling som fokuserar på hur individen uppfattar anveckling¹¹. Enligt Rivard & Huff påverkar *anvecklarens datorerfarenhet*¹² såväl uppfattning om utvecklingsverktyget som attityden till anveckling. Dessa faktorer leder i sin tur till ökad användartillfredsställelse, vilket skulle kunna göra anveckling lämpligt. Ytterligare några faktorer som kan påverka anvecklarens tillfredsställelse med anveckling är

- Tillfredsställelse med organisationens anvecklingsstöd
- Tillfredsställelse med oberoende från dataavdelningen
- Tillfredsställelse med anvecklingsmiljön
- Överensstämmelse mellan push¹³ och pull¹⁴
- Uppfattad push

Den senare faktorn (uppfattad push) påverkas av

- Dataavdelningens förändringsbenägenhet (Rivard & Huff 1988:557)

Enligt Rivard & Huff är alltså anvecklarens datorerfarenhet, dataavdelningens inställning och de miljömässiga betingelserna för anveckling viktiga faktorer för anvecklarens tillfredsställelse med möjligheterna till anveckling. Dessa faktorer skulle följaktligen göra anveckling lämplig när den är berättigad.

Studier i D1 (Avdic 1995a:99f) visar att anvecklare uppfattar stöd från såväl arbetskamrater som arbetsledning som viktiga. Dess utom redovisas att förhållandet mellan anvecklare snarast förbättrats sedan anvecklarna börjat utveckla mer aktivt.

Enligt Andersen (1994) är viktiga förutsättningar för att lyckas med vad han kallar användarledd systemutveckling:

- att användarna väljer uppgifter som lämpar sig för den här typen av systemutveckling
- att användarna har förstått och accepterat det ansvar som är förbundet med den här typen av systemutveckling
- att infocentret har rätt medarbetare med rätt attityder
- att användarna får rätt slutanvändarverktyg
- att ADB-avdelningen stöttar den här typen av systemutveckling

Andersen (1994:443)

Och vidare att

- anvecklare blir mer ekonomiskt medvetna
- anvecklare är mer benägna att engagera sig i utvecklingsarbete

¹¹ User Development of computer Applications (UDA).

¹² "Computer background".

¹³ Push betyder dataavdelningens stöd (DP department supply).

¹⁴ Pull betyder användarens efterfrågan (user demand).

Anveckling

(Andersen 1994:439)

Om det anvecklade systemet hämtar indata från en databas bör det vara fråga om en *väldefinierad och välunderhållen databas*. Det anvecklade systemet får inte uppdatera data i en företagsgemensam databas. De anvecklade systemen bör enligt Reynolds (1992) inte vara verksamhetskritiska vad gäller driftsäkerhet och de bör inte ha för höga krav på svarstid. Om det anvecklade systemet har en egen databas bör den inte vara för stor (250.000 poster) och den bör inte vara sammansatt av data från olika källor (Reynolds 1992:348). Även Hicks Jr (1990:203) poängterar vikten av en ändamålsenlig databasadministration.

Ett forskningsprojekt med näraliggande inriktning relativt KPA, är AMIS-projektet.¹⁵ (Agnér Sigbo 1993) Projektet fokuserar på användardeltagande i vidareutveckling av system. Enligt Agnér Sigbo blir inte system färdigutvecklade. Systemutveckling sker kontinuerligt, vilket överensstämmer med egenskapen *stegvis förfining* som utmärker KPA. Olika fall av systemutveckling kräver olika typ av utvecklingsarbete och Agnér Sigbo rekommenderar därför inget 'recept' utan en 'kostcirkel'. I AMIS-projektet utgår man från att människor har behov av autonomi och egenkontroll och behov av gemenskap och tillhörighet. Människor har manuella, intellektuella och estetiska förmågor och behov av att uttrycka dessa. Under forskningsarbetet har man i projektet bl.a. kommit fram till att bl.a. arbetsförutsättningar, demokratiskt ledarskap, självstyre och kreativt klimat underlättar kontinuerlig systemutveckling (Agnér Sigbo 1993:292ff).

Lämplighetsfaktorer

Lämplighetsfaktorerna är relaterade till medel för omständigheter kring anveckling, d.v.s.

Verktygets egenskaper

- lämplighet för anveckling (inte bara möjligt)

Anvecklarens egenskaper

- datorerfarenhet och andra relevanta kunskaper (utöver vad som gör det möjligt att anveckla)
- attityd (kreativitet och ansvarstagande)

Anvecklarens arbetssituation

- tid att anveckla
- anvecklarens kontroll över arbetssituationen
- lämpliga uppgifter (inte bara möjliga)
- självständiga uppgifter
- stöd som anvecklaren upplever sig ha från ledning och dataavdelning

Omgivning

- ledningskultur
- dataavdelningens förändringsbenägenhet

¹⁵ AMIS uttyds Ansvars- och Medverkansformer I kontinuerlig Systemutveckling.

- dataavdelningens inställning till anveckling
- tillgång till interna data
- organisation av interna data

Lämplighetsfaktorerna är orienterade mot hur anvecklingen skall gå till, d.v.s. organisation av arbetet. Den mest grundläggande lämplighetsfaktorn förefaller vara anvecklarens arbetssituation. En situation som skulle kunna motivera anveckling är en där först och främst anveckling är möjlig och berättigad. Dessutom skall verktygets egenskaper matcha anvecklarens kunskaper och attityd. Vidare skall arbetssituationen vara sådan att anvecklaren självständigt kan utnyttja tid för utförande av meningsfulla uppgifter. Anvecklaren skall stödjas (och uppleva att han stöds) av ledning och dataavdelning. Vid behov skall det finnas tillgång till lämpliga data.

Sammanställning av centrala faktorer i detta kapitel kan ses som en genomgång av forskning av förutsättningar för anveckling. Bland forskare finns skillnader i syn på vilket förhållningssätt som är det lämpligaste för dataavdelning och ledning. Skiljelinjen går kring vilken grad av styrning och kontroll som skall gälla för anveckling. Andersen vill t.ex. inte se anveckling som en lämplig aktivitet annat än då dataavdelningen inte har möjlighet. Synsättet kan bero på en önskan att fördela arbete mellan specialister. Denna syn står i kontrast till anvecklingsverksamhet som strävar efter integration snarare än uppdelning. Då faktorerna jämförs med de empiriska studier som genomfördes i D1 finns stora likheter i uppfattning om vilka förutsättningar som gäller.

10.4 Effekter av anveckling

Att anveckling sprids i organisationer beror bl.a. på en förväntning om positiva effekter. Att anveckling och KPA kan leda till såväl positiva som negativa effekter för olika intressenter och aktörer är uppenbart. De effekter som beskrivs i detta avsnitt har kategoriserats induktivt efter analys av ett antal referenser som behandlar anveckling. Innehållet i detta avsnitt är inte kategoriskt skilt från avsnitten om spridning och förutsättningar. I viss utsträckning överlappar spridningsfaktorer, förutsättningar och effekter varandra, åtminstone är de starkt relaterade.

10.4.1 Systemoptimalitet - verksamhetskunskaper

En negativ effekt som Reynolds (1992) och Andersen (1994) anför är *suboptimerade eller icke-optimala systemlösningar*. Den orsak som man anför till denna negativa effekt är att anvecklares kunskaper om och förståelse för verksamheten är otillräckliga. Reynolds menar att IT-specialister kan ha större förståelse för verksamheten än anvecklare. Detta innebär i sådana fall att det kan leda till icke-optimala system. "End users (like many IS professionals) are quick to develop programs without fully understand the root of the problem or needs that must be adressed." (Reynolds 1992:353) Andersen ser ett alltför decentraliserat beslutsfattande som en tänkbar orsak till att anvecklares systemlösningar ej blir av "strategisk karaktär". Andersen menar att det finns en risk att "...för lite av utvecklingsresurserna används till system som ger marknadsvinster." (Andersen 1994:445)

Att det är verksamhetsföreträdarens kunskaper om verksamheten som är en av de mest grundläggande mekanismerna bakom KPA diskuteras i termer av verksamhetsmotverkande chauvinism i kapitel 1. Det är dock långtifrån självklart att KPA leder till optimala system. Mikas redogörelse i kapitel 5 styrker detta. Diskussionen om icke-optimala system kan också i viss mån vara en produkt av en Livscykelnsyn, där informationssystem är komplexa fenomen som kräver stora resurser vid såväl planering som utveckling och drift/förvaltning. Flera av de beskrivna systemen i *Del II* har inte utvecklats i ett sammanhang och har inte varit optimala från början. Ett exempel är **täckningsbidrag per timme** (Se Figur 39 sid 78) som kan ses som en process för analys av produktionsmätning snarare än en system.

10.4.2 Systemkvalitet - systemutvecklingsarbete

Anvecklares agerande inom ramen för den process som formuleras av livscykelmodellen (se kap 9) anses av flera forskare orsaka låg kvalitet hos anvecklade system. Huruvida detta är en följd av en specifik uppfattning om hur systemutveckling skall gå till, en specifik uppfattning om vad som är ett informationssystem eller en i alla avseende berättigad reflexion kan diskuteras. Den låga kvaliteten kan ta sig uttryck i

- felaktiga utdata (se även kap 10.4.3)
- oflexibla system
- personberoende
- bräckliga system (låg driftssäkerhet)

Andersen ser *avsaknad av förändringsanalys* som en tänkbar orsak till låg kvalitet på system, även om de är av mindre omfattning (Andersen 1994:442). Under själva utvecklingsarbetet anges bristande testning som en orsak till den sämre systemkvalitet av Reynolds (1992:353) och Hicks Jr (1990:199). Enligt Carlsson (1993:228) är anvecklare ofta dåliga modellbyggare genom att de bryter mot 'goda' principer, vilket leder till system med låg kvalitet. En annan felkälla är enligt Hicks Jr (1990:203) dåliga indatakontroller och dålig systemsäkerhet. Alter (1996:624) ser anvecklares dåliga programmeringskunskaper som en orsak till bräckliga system. Komplexa och röriga system finns representerade i *Del II*, t.ex. berättar Sten och Mika i kapitel 5 om detta. Anvecklarnas yrkeskunnande och verksamhetsansvar ses som en faktor som motverkar dålig systemsäkerhet eftersom systemen är en del av deras arbetsuppgifter.

Dokumentation av anvecklade system är inte självklar. Lally (1995:4) menar att det är ett av kännetecknen på anvecklade system att de inte är dokumenterade. Kalkylsystem kan vara utvecklade så att de inte behöver dokumenteras utöver vad systemet i sig innehåller. Dålig dokumentation framhålls dock som en vanlig brist hos anvecklade system. Att anvecklade system kan vara otillräckligt dokumenterade eller inte alls dokumenterade kan beskrivas både som en orsak och en effekt. Dålig dokumentation kan resultera i system som är svåra att underhålla och/eller system som är personberoende (Reynolds 1992:353, Avdic 1995a:171, Hicks Jr 1990:192). Bland de studerade systemen var dokumentation sällsynt.

Hicks Jr ser dålig *styrning* som huvudorsak till redundans, dålig dokumentation, dålig systemsäkerhet, dåligt testade system, personberoende och dåliga indatakontroller (Hicks Jr 1990:199). Alter (1996) ser det som ett problem att anvecklade system kan behöva *kontroll och underhåll* från dataavdelningen (Alter 1996:624). Orsaken är att detta leder till att utvecklaren blir beroende av dataavdelningen, vilket kan uppfattas som ett problem. Denna syn kan ses som ett uttryck för ett synsätt som inte förespråkar hög kontroll vad gäller anveckling. I slutet på detta kapitel sammanfattas aspekter på styrning och kontroll.

Ett problem som gäller anvecklade kalkylsystem¹⁶ är *versionsbyten* (se kap 5.2). När det gäller t.ex. MS Excel kan versionerna komma med 18-24 månaders mellanrum. Detta kan få flera såväl negativa som positiva effekter. En negativ effekt kan bli att gamla versioner av kalkylsystem kan vara svåra att köra i den nya versionen av programmet p.g.a. kompatibilitetsproblem (se även resursslöseri kap 10.4.4). Utvecklarens ovana vid den nya versionen kan också skapa problem. En positiv effekt av versionsbyten är att ny funktionalitet kan bli tillgänglig för utvecklaren vilket kan göra det möjligt att utföra arbetsuppgifter som tidigare inte var möjliga, vilket kan öka systemens funktionalitet och kvalitet. Eftersom olika versioner av program kan försvåra kommunikation mellan personer i en organisation, kan ledningen tvingas styra när versionsbyten skall tillåtas, vilket beskrivs av Sten och Hans i kapitel 5.

10.4.3 Informationskvalitet

Ett informationssystem producerar information med hög kvalitet då informationen är relevant, korrekt och avpassad efter informationsmottagaren på så vis, att den endast når de avsedda mottagarna på ett begripligt sätt. Det yttersta kriteriet på hög informationskvalitet är att det bidrar till att uppfylla verksamhetens (och samhällets) mål. Informationssystem är till för att producera efterfrågad information. Av detta skäl kan man säga att systemkvalitet kan vara ett sätt att uppnå informationskvalitet, vilket innebär att diskussionen om systemkvalitet ovan indirekt även gäller för informationskvalitet. Orsaker som direkt påverkar informationens kvalitet är sådana som orsakar felaktiga utdata. Till dessa orsaker kan räknas dålig testning, som kan leda till felaktig bearbetning, som i sin tur kan leda till felaktiga utdata. På samma sätt kan anföras att dålig dokumentation kan leda till felaktig hantering av systemet som eventuellt kan leda till felaktiga utdata.

Otillräcklig *systemintegration* ses av Reynolds (1992:348) som ett problem, som kan leda till flera registreringar av samma data, vilket i sin tur kan leda till inkonsistenta system, vilket av systemutvecklingsspecialister ofta anses som en icke önskvärd effekt. På samma sätt kan dubblade system leda till redundans, vilket på samma sätt kan leda till inkonsistenta system (systemdubbling kan även betraktas som resursslöseri, se kap 10.4.4).

¹⁶ Och många andra anvecklade system.

10.4.4 Resursanvändning

Problemen, som beskrivs ovan, handlar i flera fall indirekt om resursanvändning och därigenom om ekonomi. Få verksamheter, om någon, kan undvika att ta hänsyn till direkta eller indirekta ekonomiska effekter. Ett sätt att bedöma *ekonomiska effekter* av anveckling är att värdera kostnaderna för själva anvecklingsprocessen. I en studie visar Gable, Yap & Eng (1991) att chefer och högre tjänstemän inte är medvetna om vilken kostnad som är förknippad med utveckling av kalkylsystem (anveckling). Studien beräknar kostnaden för utveckling med måtten timlön för nedlagd tid. Med denna metod blir kostnaden för anveckling betydligt högre då den utförs av chefer än av andra anställda. Anvecklarna visade sig ej känna till hur mycket tid som nedlagts på anveckling. Den nedlagda tiden var högre än vad anvecklarna själva hade trott. Mätmetoden tog ej hänsyn till ev. kvalitetseffekter och huruvida anvecklarna upplevt andra effekter av anvecklingen, som t.ex. djupare problemförståelse. Inte heller diskuterades huruvida det varit möjligt att låta t.ex. en programmerare utföra utvecklingsarbetet med likvärdigt resultat. Att låta anvecklare utan systemutvecklingsutbildning eller –erfarenhet utveckla system kan enligt Andersen (1994:445) uppfattas som ett oklokt sätt att utnyttja befintliga resurser.

Resursslöseri kan även gälla uppgraderingar av programvaror. Det kan bli kostsamt för ett företag att uppgradera till en ny version av ett program. För det första kostar programvaran pengar, för det andra kan det bli nödvändigt att uppgradera maskinvaran och för det tredje kan utbildning på den nya programvaran bli aktuell, vilket dels förorsakar en kostnad i sig samtidigt som arbetstid går förlorad. Dubbellagrade data och dubblade system kan (som visats ovan) ses som ett hot mot informationskvaliteten, men också som ett uttryck för resursslöseri (Hicks Jr 1990:199, Reynolds 1992:353).

Klepper (1990:5ff) presenterar en kostnads/intäktsanalys för lönsamhetsberäkning av anveckling. Beräkningen utgår från ett val mellan att utveckla system inom en given avdelning eller genom att anlita någon som arbetar utanför den egna avdelningen. Särskilt uppmärksammas sk transaktionskostnader, vilka uppstår vid systemutvecklingsprojekt genom att man måste förhandla, styra och genomdriva arrangemang som inbegriper de inblandade parterna i ett projekt. Parametrar som används för val av utvecklingsstrategi (anveckling eller TSU) är enligt Klepper:

- **Relativ kostnad.** Kostnaden relateras till systemstorlek, utvecklingsansats och användarerfarenhet.
- **Tillgångsspecifitet.**¹⁷ Förhandlings- och genomdrivningskostnader ökar ju mer specifik en tillgång är. Om det är få användare är kostnaderna lägre.
- **Osäkerhet.**¹⁸ Uppstår beroende på verksamhetens sammanhang och utifrån dataavdelningens systemutvecklingskultur.
- **Kontaktfrekvens.** Rikliga och goda kontakter minskar kostnaden.

Flera av de faktorer som nämnts kan delvis relateras till tidsåtgång. Att anveckling resulterar i *lägre tidsåtgång* är ett argument som inte sällan återkommer. T.ex. nämner

¹⁷ "Asset specificity". Hur pass specifik en tillgång är.

¹⁸ Vad gäller beslut om informationssystemets utveckling.

Alter (1996:624) och Robey & Zmud (1990:15) att anveckling spar tid och att TSU ger långa ledtider. Enligt Robey & Zmud leder anveckling inte bara till sparad tid utan även till *ökad individuell och organisatorisk produktivitet*.

I förlängning av de kritiska resonemang om resursanvändning och anveckling, som framförs ovan ligger ett resonemang om att specialisering och fördelning av arbetsuppgifter är önskvärd för att hushålla med resurser. Denna synpunkt är rimlig i de allra flesta sammanhang men står i viss motsats till anveckling som fenomen där integration av arbetsuppgifter snarare är det typiska. I ett anvecklingssammanhang är utvecklarens verksamhetskunskap det väsentliga. Utifrån denna verksamhetskunskap kan utvecklaren förse sig med kunskaper ur olika kunskapsdomäner för att förverkliga sina intentioner, om han anser det befogat även om det inte är det mest resurseffektiva. I Avdic (1995a) visas att utvecklare inte ser systemutveckling av IT-specialister som ett alternativ till deras anvecklingsverksamhet. Huruvida detta är ett resultat av lönsamhetsbedömningar eller ej framgår ej av studien.

10.4.5 Beslutsfattande och arbetssituation

Omedvetna förändringar av organisationers informationspolitik kan vara en effekt av anveckling enligt Carlsson (1993:228) dessutom finns en risk för *obalans mellan kontinuerlig utveckling och förändring av beslutsprocesser*. Carlsson diskuterar också påverkan på individen och individuellt lärande¹⁹. Carlsson visar i en studie att det skedde ett lärande som visade sig genom förändringar av uppgifter, aktiviteter och uppgifters ändamål samt användarnas mentala modeller av informationskällorna. Beslutsstödsystemet i Carlssons studie (kalkylsystemet) var medlet för lärandet.

I en studie om förändringar av kontakter som följer med ökning av datakommunikation menar Séror (1990:105ff) att *andelen direkt kommunikation minskar*. Andra effekter som uppstår i en ”rikare informationsmiljö” är *ökad arbetstakt och kortare informationscykler*.

En tänkbar positiv effekt av anveckling²⁰ skulle kunna vara *mer berikat arbete*²¹. Enligt Kling, Iacono & George (1990) förväntar sig arbetsledare och chefer *förbättrade prestationer*²². Särskilt vanligt var det att professionella²³ upplevde denna förväntan på förbättrade prestationer (Kling, Iacono & George 1990:63ff).

Enligt Smithson & Hirschheim (1990:37ff) skapar anveckling möjligheter att förändra sociala relationer genom att organisationshierarkier bryts ned. Anveckling kan öka variationen i arbetsuppgifter och kvalifikationer²⁴. Anveckling kan även förbättra återkoppling i arbetet. Dessa faktorer kan leda till uppfattad högre status i arbetet och ökad

¹⁹ "individual learning".

²⁰ "desktop use".

²¹ "increased job enrichment".

²² "improved performance".

²³ "Professionals".

²⁴ "Skill requirements".

arbetstillfredsställelse. I ett samhälleligt perspektiv kan anveckling påverka möjligheten till hälsa, utbildning, välfärd, att få jobb etc..

Att anveckling medför förändring av villkor för arbetet stöds av t.ex. Ismo (se kap 3.3) och Jan (se kap 4.4). Betydelsen av nätverk i anvecklingsmiljön stöds av såväl Avdic (1995a:80;100;117) som av studierna Industri, IT-specialister och Applikation. Hur anvecklare upplever sin egen arbetssituation och status har inte behandlats i D1 eller D2.

10.4.6 Inflytande och styrning

Synen på (behovet av) styrning och kontroll är varierande mellan forskare. Avsaknad av standards och styrning kan leda till mindre framgångsrik anveckling menar Hicks Jr (1990:199). Ett föremål för styrning är databasadministration, som kan vara en viktig förutsättning för anvecklares tillgång till information (Hicks Jr 1990:203). Smithson & Hirschheim (1990) menar att styrning kan bero på oro över att anvecklare lägger ner för mycket tid på att utveckla olämpliga system.

"The concern with control may reflect a genuine worry on the part of an organization's management that users may waste much time developing systems that are unnecessary, incompatible, inaccurate, unreliable, and unmaintainable."

(Smithson & Hirschheim 1990:37ff)

Ett annat uppmärksammat problem är den förändring i status och inflytande hos IT-specialister, som kan bli effekten av utbredd anveckling "The IS department loses both status and power." (Smithson & Hirschheim 1990:37ff)

En motvikt till de farhågor som nämns ovan är en syn som ser anveckling som positiv med avseende på det inflytande som anvecklaren får. Alter (1996) ser det faktum att anvecklaren kontrollerar applikationen som en fördel eftersom han då kan förändra den om så skulle behövas. Alter (1996:624) Westin (1993) samt Greve & Karlsson (1994) noterar att en effekt av anveckling kan vara bättre *styrning* av verksamheter samt lokalt ekonomiskt inflytande. Westin (1993) behandlar i en studie utveckling av lokalt informationsstöd (LIS). Ett antal LIS utvecklades med hjälp av kalkylprogram för att "...nä den relevans, flexibilitet och snabbhet som krävs för aktivitetsbaserad²⁵ styrning" (Westin 1993:sista sidan). Två slutsatser från studien är

- a) centrala system har svårt att leva upp mot ett aktivitetsbaserat krav på styrprocessen (Westin 1993:222) och
- b) "...utveckling och användning av lokala informationsstöd ger ett antal positiva effekter som skärper styrning, ökar engagemang och lärande" (Westin 1993:368).

Westin konstaterar vidare "Egen design är en nödvändig ingående del för utveckling av LIS. Inga större svårigheter att utveckla kompetens hos arbetsledare som utvecklade och nyttjade LIS vad gäller datahantering förelåg. ... LIS fungerar bäst under eget ansvar och engagemang. Användning fordrar ledningens acceptans och stöd" (Westin 1993:247).

²⁵ Aktivitetsbaserade data är "...data som fångas i direkt anslutning till aktiviteter" (Westin 1993:227).

Greve & Karlsson (1994) berör användning av kalkylprogram vid en studie kallad *Att utöva inflytande med lokala informationssystem*. Inom ramen för studien ges undervisning i ekonomi och kalkylprogramanvändning till ett antal handläggare. Detta resulterar i att handläggarna så småningom i egen regi bygger upp ett antal budgetmodeller för styrning av den egna verksamheten. Studien visar att kalkylprogram kan bidra till möjligheter att utveckla lokala informationssystem i anvecklares egen regi och att dessa system ger anvecklare större möjligheter till styrning av den egna verksamheten.

Styrning för anveckling kan enligt ovanstående resonemang utövas på åtminstone två plan, dels vad gäller organisatoriska förutsättningar, t.ex. databasorganisation och tillgängliga programvaror, dels vad gäller anvecklarens direkta arbete, t.ex. om anvecklaren skall få utveckla egna system eller om han skall använda specificerade standardmoduler. Nedan diskuteras frågan om styrning vidare.

10.4.7 Summering effekter - styrning eller ej

Mänsklig verksamhet tenderar till att bli alltmer komplex. För att hantera tilltagande komplexitet i organisationer kan två grundläggande mekanismer identifieras: arbetsfördelning och koordination (Mintzberg 1983:3). Inom informatikområdet sker en specialisering i form av uppdelning i olika kompetensgrupper. Specialisering kan ske på olika typer av arbetsuppgifter, t.ex. analys, databasdesign eller programmering. Specialiseringen kan ske inom en arbetsuppgiftsklass, t.ex. programmering i C++, Cobol eller Visual Basic. Då specialisering ökar, ökar även behovet av koordination. Mintzberg har identifierat fem grundläggande typer av koordination: 1) ömsesidig anpassning, 2) direkt styrning, 3) standardisering av arbetsprocesser, 4) standardisering av "output" samt 5) standardisering av kunnande (Mintzberg 1983:7). De studerade anvecklarna i kapitel 3 & 4 samt de flesta i delprojekt D1, kan sägas vara föremål för koordinationsformen *ömsesidig anpassning*. I själva verket förefaller denna den mildaste av koordinationsformer vara en förutsättning för anveckling i den form som presenteras i denna avhandling. Medan systemutvecklingsområdet i övrigt kan sägas vara en produkt av en önskan att fördela arbetsuppgifter, står anveckling som fenomen i motsats till fördelningsambitionen genom att istället sträva efter integration av specialistkunnande. Nu är denna ansats inte en oväntad anomali i Livscykelparadigmet. I själva verket har de rationaliseringsbehov som på sätt och vis drivit systemutvecklingsämnet framåt kommit att omfatta det egna ämnet. Viss systemutvecklingskunskap kan formaliseras och göras tillgänglig för icke-IT-specialister, t.ex. i form av kalkylprogram, vilket har möjliggjort att systemutveckling kan bedrivas i viss omfattning av andra än IT-specialister. Ur diskussionen om effekter av anveckling i kapitlet ovan, kan urskiljas två delvis motsatta synsätt. Det första synsättet, som förespråkar hög kontroll, ser anveckling som en potentiell risk om det inte sker en noggrann styrning. Det andra synsättet, som inte förespråkar hög kontroll, ser anveckling som en möjlighet, som inte skall styras för hårt. Denna syn kan ses som ett uttryck för ömsesidig anpassning. Det första synsättet, *hög-kontroll-synsättet*, ser systemutvecklingsspecialisterna som de som skall utföra systemutveckling i ett företag medan användarna skall vara just användare och eventuellt göra mindre enskilda system. Det andra synsättet,

Anveckling

låg-kontroll-synsättet, ser dataavdelningen som en grupp personer som håller fast vid sin specialistfunktion och status, trots att andra möjligheter kan finnas för handen.

Som representant för *hög-kontroll-synsättet* skulle Andersen (1994) kunna väljas. Andersen anser inte att användare skall utföra IT-specialistarbete. Enligt Andersen bör ”användarledd systemutveckling” vara ett komplement till den traditionella systemutvecklingen där ADB-experten deltar. Andersen förordar att användarledd systemutveckling styrs via ett infocenter. Andersen menar att:

...det räcker inte med att erbjuda användarna ett verktyg. Det måste dessutom finnas riktlinjer för hur användarna skall arbeta, det vill säga en systemutvecklingsmodell. Modellen måste vara mycket flexibel. Uppgiftens art måste bestämma vad användarna bör använda av modellen. Ska de enbart göra en extra rapport i ett existerande system, är det inte nödvändigt att utföra något analysarbete. Men när de skall göra en fullständig, om än liten, tillämpning måste utvecklingsmodellen uppmana användarna att inleda arbetet med förändringsanalys (Andersen 1994:441f).

Smithson & Hirschheim (1990) kan sägas vara representanter för *låg-kontroll-synsättet*. I följande citat argumenterar Smithson & Hirschheim mot *hög-kontroll-synsättet*.

The implicit assumption seems to be that end-users are untrained programmers who, given the opportunity, will enthusiastically deluge the organization in ‘spaghetti’ code, running on incompatible machines, in the same way that professionals did until recently. Thus, it seems to follow that they should be controlled and trained. Whilst there may be a few isolated instances of this type of behavior ... we would regard this as a rather one-sided and in some ways dated view (Smithson & Hirschheim 1990:38).

Ytterligare en aspekt av synen på anveckling ges av Nurminen (1988) (se kap 8.4) som ser anveckling som uttryck för ett annat paradigm än det förhärskande (dominerat av TSU-tänkande). I detta olika paradigm kan inte samma måttstockar tillämpas.

En tänkbar förklaring till *hög-kontroll-synsättet* kan vara att anveckling betraktas som en kvalitativt likvärdig sysselsättning jämfört med TSU. Andersen (1994) ser användaren som den som kan specificera ”systemets yttre egenskaper”. Användaren är en, visserligen mycket viktig och central, men ändå kugge i ett systemutvecklingsprojekt. Då det gäller systemutvecklare är det naturligt att styra dessas verksamhet i och med att det är deras huvudverksamhet. När det gäller redovisare, produktionsplanerare, kvalitetsansvariga, arbetsledare etc. har dessa en annan huvuduppgift som styr deras roll i organisationen. Den syn som denna avhandling strävar efter att framföra är att anveckling bara är möjligt om styrningen inte är för stark.

10.5 Arbetssätt

Detta avsnitt syftar till att redogöra för olika uppfattningar om hur anveckling går till och bör gå till. Inledningsvis ges en beskrivning av det arbetssätt som kännetecknar KPA enligt Avdic (1995a), dvs D1:

- Stegvis förfining

- Induktivt arbetssätt
- Avsaknad av standardiserade metoder
- Sammanflätade aktiviteter
- Implicit förändringsanalys
- Strävan efter oberoende från systemutvecklingsspecialister (Avdic 1995a:179ff)

Stegvis förfining innebär att kalkylsystemet utvecklas gradvis och att det ofta kan användas under hela utvecklingsprocessen. Utvecklingsprocessen kan variera från någon minut till flera år. KPA är följaktligen inte en projektorienterad aktivitet.

Ett *induktivt arbetssätt* innebär att utvecklaren utgår från arbetsuppgiften och låter utvecklingsarbetet utformas utifrån detta. Arbetssättet är *inte deklarativt* eftersom utvecklaren inte beskriver hur data skall organiseras på en metanivå på det sätt som ibland görs i databashanteringsprogram. Arbetssättet är *inte heller prototypingorienterat* i den meningen att man gör en modell som så småningom skall utvecklas till det färdiga systemet. I KPA kan systemet mycket väl användas från början. Arbetssättet är *inte heller procedurellt* på det sätt som då systemutveckling bedrivs med programmeringsverktyg. Normalt används *inte* programmering i KPA. Ett undantag från ovanstående är kalkylsystem av typen *Applikationen*, vilket behandlas närmare i kapitel 6.

Avsaknad av standardiserade metoder innebär att KPA sällan utförs på något formaliserat sätt, utan drivs av de krav som arbetsuppgifter ställer.

Sammanflätade aktiviteter innebär att användning, utveckling och lärande sker parallellt för utvecklaren.

Implicit förändringsanalys innebär att ingen specifik förändringsanalys utförs innan utveckling påbörjas utan att den utförs integrerat med utvecklingsarbetet.

KPA kan betraktas som en *strävan efter oberoende* från IT-specialister. Detta innebär inte att det är frågan om en medveten opposition mot dataavdelningarna utan snarare en önskan att självständigt kunna utnyttja datorer på verksamhetens villkor.

10.5.1 Traditionalism, organisation eller förnyelse?

En motsägelsefull situation råder vad gäller KPA, eftersom ovanstående kännetecken kan sägas stå i viss motsättning till den metodtradition som traditionellt råder inom systemutveckling (se kap 9). I litteratur om anveckling kan skönjas en önskan att förmedla riktlinjer (styrning) för anveckling. En typ av riktlinjer ansluter till Livscykelmodellen (se kap 9), dvs man rekommenderar att anveckling i stort skall ske enligt faserna analys, design, utformning och implementering (Panko 1988:49ff, Mähring 1989, Benham, Delaney & Luzy 1993). Enligt en annan syn ses en organisatorisk hantering av anveckling som en lösning. Denna lösning kan anta formen av ett sk Information Center (IC) (Flensburg 1986, Capron 1996). Denna syn kan innebära att man ser anveckling som utveckling av enbart små system och som därför inte kräver någon mer omfattande systematik (Andersen 1994:437ff, Alter 1996:38). En tredje syn

skulle man kunna kalla de ansatser som söker andra vägar än de två som nämnts (Hicks Jr 1990:187ff, Isakovitz, Shocken & Lucas Jr 1995). Gränserna mellan synsätten är flytande. Vad som t.ex. är att betrakta som ett litet system, kan variera på samma sätt som uppfattningen om vad som är TSU. Vissa forskare, t.ex. Panko diskuterar såväl efterföljande av livscykelmodellen som användning av IC. Någon form av stöd får alla anvecklare i organisationer, även om det inte kallas IC. Det väsentliga är att det första synsättet (*traditionalism*) vill överföra befintliga metodtraditioner på nya former av systemutveckling. Det andra synsättet (*organisation*) fokuserar på organisatoriskt styrande av anveckling. Det tredje synsättet (*förnyelse*) är inte en homogen samling av synsätt på anveckling. Som en gemensam nämnare för dessa synsätt uppfattas en syn att anveckling är något kvalitativt annorlunda än systemutveckling. Det tredje synsättet accepterar anveckling som systemutveckling samtidigt som traditionella metoder bedöms vara icke ändamålsenliga i sin helhet. Det tredje synsättet framhåller inte organisatorisk styrning av anveckling som det främsta alternativet för hantering av anveckling i organisationen. Denna avhandling kan ses som ett uttryck för förnyelsesynsättet. Beskrivningen av de tre synsätten kan synas hårddragen men ambitionen är att urskilja tendenser i synsätt snarare än att beskriva enskilda uppfattningar. Valet av benämningar har främst en klassificerande och beskrivande ambition (se Figur 84 nedan).

<p>Traditionalism: Anveckling behandlas som TSU</p>	<p>Organisation: Anveckling stöds genom IC eller liknande</p>	<p>Förnyelse: Anveckling kan användas på andra system än mindre, TSU är inte ändamålsenlig</p>
---	---	--

Figur 84 Olika syn på arbetssätt för anveckling.

10.5.2 Traditionalismsynsätt

Som representanter för traditionalism räknas här Mähring (1989) och Panko (1988)²⁶. Det kan noteras att referenserna är ca tio år gamla, vilket kan tolkas som att det låg närmare till hands att applicera traditionella synsätt för tio år sedan än vad det gör i dag (1999). Att traditionalism ändå tas med sker inte bara av historiska skäl, metodtraditionen inom systemutvecklingsområdet kan knappast undvikas då diskussion förs om arbetssätt för systemutveckling diskuteras, även om det gäller anveckling.

Panko (1988:49ff) beskriver utveckling i ett anvecklingsssammanhang som bestående av faserna Assessment, Design, Construction och Implementation. Med Assessment menas hur lämpliga anvecklingsprojekt värderas och väljs. Design innebär planering av systemet. Construction innebär byggandet av applikationen. Till byggande räknas inte bara programmering, testning och dokumentation utan även inköp av maskinvara, programvara samt datakommunikationsutrustning. Implementation innebär att det färdiga systemet infogas i den befintliga arbetsorganisationen. Faserna i Pankos modell kan delas in i aktiviteter. I Designfasen utförs t.ex. målformulering, förståelse av den befintliga arbetsorganisationen, identifiering av användarbehov samt, som sista aktivitet design av det nya systemet. Pankos modell har mycket gemensamt med Livscy-

²⁶ Pankos bok End User Computing (1988) har en beskrivande karaktär och tar även upp synsätt som kan räknas till de andra skolorna.

kelmodellen (se kap 9) och fungerar även på detta sätt med den skillnaden att användare är huvudaktörer i utvecklingsarbetet. Eftersom utvecklingsarbetet drivs av en avdelning, spelar avdelningsorganisationen roll för utvecklingsarbetet. Avdelningschefsrollen presenteras av Panko som den mest centrala i utvecklingsarbetet. Övriga roller av betydelse för utvecklingsarbetet är sekreterare, kärnpersonal²⁷ och s.k. ”power users”. Utvecklingen av ett system beskrivs som en projektliknande aktivitet där ”power users” har en IT-specialistliknande status. Enligt Pankos modell är inte befintliga verktyg en självklar utgångspunkt för val av utvecklingsmiljö, vilket innebär att parallellerna till KPA inte är så starka.

Panko & Halverson Jr (1996)²⁸, har speciellt studerat anveckling med kalkylprogram och antyder att en orsak till felaktiga kalkylsystem är bristen på inledande analys. Panko & Halverson Jr menar att utvecklare bör ta efter IT-specialisters tillvägagångssätt för att minska felfrekvensen i kalkylsystem. Förslag till åtgärder som presenteras är :

- planering av kalkylsystemet före konstruktion
- testning av kalkylsystemet före användning
- användning av cellskydd
- utformning av kontrollpolicy för organisationen

Detta förslag till utvecklarpaxis kallar Panko (1996) att ”lära nya hundar gamla konst²⁹”.

Mähring (1989) har utifrån ett praktikfall (budgetsystem vid Handelshögskolan i Stockholm) formulerat en systemeringsmetodik för att konstruera kalkylmodeller³⁰. Mährings arbete är intressant därför att det är ett exempel på ett försök att formulera en heltäckande metod för utveckling av just kalkylsystem. Den strukturerade metodiken som Mähring rekommenderar består av:

- verksamhetsanalys
- informationsstudie (informationsprecedensanalys och datamodellering)
- systemutformning

Mähring rekommenderar användande av metodik för utformning av kalkylmodeller. Ett viktigt skäl till detta är att problemen med utveckling av kalkylmodeller ökar med modellernas komplexitet. Förslag på lösningar är:

- grafisk formelrepresentation
- hjälpprogram för dokumentation och felsökning
- strukturerat programmeringsspråk
- tillämpning av top-down programmering
- tumregler för verifiering av kalkylmodeller.

Noteras kan att kalkylsystemet som utvecklades, och låg till grund för studien, framstår som relativt komplext (av typen *Stora kalkylsystemet*). Modellen tycks i viss utsträckning ha utvecklats av någon/några person(er) till några andra, vilket tyder på att

²⁷ ”Staff professionals”.

²⁸ Även Andersen (1994:391), Capron (1996:391) och Mähring (1989).

²⁹ ”Teaching new dogs old tricks”.

³⁰ Mähring använder begreppet kalkylmodeller, vilket är delvis synonymt med kalkylsystem. Min tolkning är att man vid tiden för uppsatsens skrivande (1989) övervägande utnyttjade kalkylprogram för att konstruera just kalkylmodeller.

kalkylprogrammet använts som ett TSU-verktyg snarare än ett anvecklingsverktyg. Verktyget tycks även ha haft en relativt begränsad funktionalitet jämfört med kalkylprogram som är av tio år yngre modell. De tre första förslagen på lösningar (grafisk formelrepresentation etc.) förekommer idag (1999) i kalkylprogram. Den utökade funktionaliteten hos kalkylprogram innebär att en del funktioner som krävde programmering 1989 idag kan realiseras med hjälp av inbyggda funktioner. Exempel på detta är rapportdesign, uppdelning av modeller i logiska enheter och statistisk analys.

10.5.3 Organisationssynsätt

Organisationssynsättet innebär att stöd till anvecklare ges i form av organisation av verksamheten snarare än genom metodik. Som representant för organisationssynsättet räknas här Flensburg (1986), Capron (1996), Andersen (1994), Hicks Jr (1990) och Alter (1996). En förekommande variant av organisation som föreslås är Information Center (IC). Idén med IC har diskuterats av fler än nämnda författare. De flesta större verk (i synnerhet på 1980-talet) som berör anveckling behandlar IC på ett eller annat sätt. På senare tid har idén med IC inte behandlats lika omfattande. Viss kritik har riktats mot idén med IC, bl.a. av Brancheau & Brown (1993:455) som framhåller att mer informella former av stöd upplevs som effektivare av användare. Även Robey & Zmud (1990:18) menar att IC-funktionen har begränsningar.

1986 publicerade Per Flensburg sin doktorsavhandling under titeln *Personlig databehandling*. Flensburg beskriver ett fenomen som under första hälften av 80-talet betraktades som nytt, nämligen användare som bygger informationssystem på egen hand. Flensburg beskriver två typfall det ena från ett stort, det andra från ett litet företag. Det stora företaget har ett IC. Introduktionen av utvecklingsverktyg på företagen beskriver Flensburg som ett väsentligt val, och skissar på en tänkbar arbetsgång för introduktion av personlig databehandling. Arbetsgången när det gäller införandet av utvecklingsverktyg bygger till stor del på den syn på arbetsmetodik som livscykelmodellen (se kap 9) ger uttryck för dvs:

- Problemanalys
- Informationsanalys
- Utformning
- Implementering.

Då verktyg för användare valts, skall stöd ges genom en grupp personer som Flensburg kallar för mellanhänder. Enligt Flensburg kan organisation av experters användarstöd ske genom inrättande av mellanhänder och kontaktpersoner.

För vissa företrädare av organisationssynsättet uppfattas inte anveckling som egentlig systemutveckling. Definitionen av systemutveckling är att det är TSU. En följd av detta är att systemutveckling skall bedrivas av IT-specialister för användare. Ett exempel på denna syn visar Alter (1996:38) som menar att anveckling endast är effektiv för små system där IT-specialister inte behövs. Andersen (1994:351) menar att användarledd systemutveckling³¹ i praktiken bör begränsas till mindre ADB-system och inte

³¹ Synonymt med EUC enligt tidigare definition.

kan omfatta centrala system. Ett IC³² framhålls som ett sätt att organisera denna typ av verksamhet. Enligt Andersen krävs noggrann organisation för lyckad användarledd systemutveckling. Andersen föreslår att IC organisatoriskt placeras under ADB-chefen på samma nivå som SU-avdelningen och driftsavdelningen. En liknande lösning presenteras av Hicks Jr (1990:200f).

Ett skäl till att företag kan välja denna typ av strategi är, enligt Andersen, att systemutvecklarna har en tung arbetsbörda. Andersen skriver vidare om användarledd systemutveckling: "I sin yttersta form är denna strategi inte särskild ändamålsenlig eftersom den på sätt och vis bortser ifrån att det finns en yrkesgrupp (systemutvecklare) som kan hjälpa till med systemutvecklingen och bidra till en bättre lösning" (Andersen 1994:351). Användarledd systemutveckling stöttar uteslutande utvecklingsfasen av ett ADB-system, poängterar Andersen. Han menar att verktygen styr arbetet och att de inte har några recept på hur *analysarbetet* skall utföras. Det kan till och med vara så att verktygets egenskaper gör att analysen begränsas till att endast behandla uppgifter som erfordras till ett speciellt verktyg. Andersens syn på anveckling är försiktig. Han är avvaktande positiv till vissa former som innefattar starka krav på styrning och organisation. De risker som Andersen förknippar med anveckling, beskrivs i kapitel 10.4.

10.5.4 Förnyelsesynsätt

En reflexion som skulle kunna göras efter genomgången ovan är att få metodansatser har uttalat utformats utifrån de kännetecken som utmärker anvecklingsaktiviteter (t.ex. KPA). Resultatet från Brancheau & Browns (1993) studie över forskning om anveckling tyder t.ex. på att styrda och organiserade aktiviteter kan vara mindre effektiva än informella. Rubriken *förnyelsesynsätt* strävar efter att samla synsätt som inte huvudsakligen strävar efter att inordna anveckling efter de andra två synsätten. Ett exempel på detta är Isakovitz, Shocken & Lucas Jr (1995) samt Isaksson (1996) och Avdic (1997a, 1997b).

Isakovitz, Shocken & Lucas Jr (1995) har utformat en alternativ modell för utvecklande av kalkylsystem utifrån fyra komponenter som man menar utmärker alla kalkylsystem. De fyra komponenterna är: *schema, data, editorial, and binding*. Man presenterar en sk faktoriseringsalgoritm för att identifiera och extrahera komponenterna samt en syntetiseringsalgoritm som skall kunna hjälpa användare att konstruera exekverbara kalkylsystem utifrån återanvändbara modellkomponenter. Syftet sägs vara att stödja konstruktion, delning och återanvändning av kalkylsystem i organisationer.

En annan ansats som utgår från det specifika med utvecklingsverktyget kan kallas *den detaljerade tumregelsansatsen* (DDT). DDT innebär att specifika regler och råd utformas för specifika behov i motsats till generella ansatser som har ambitionen att täcka många problem i många miljöer. Exempel på DDT är Isaksson (1996). Denna ansats ger råd om hur man skapar en viss sorts diagram, hur man utför en investeringskalkyl eller hur man kopplar en tabell till en dialog. Isaksson berör överhuvudtaget inte

³² Andersen använder uttrycket Infocenter.

Anveckling

metodaspekter på annat sätt än genom tumregler för att utföra specifika funktioner eller för att lösa vissa typer av problem.

Ytterligare en tumregelsansats som utgår från studiet av KPA är Avdic (1995a, 1997a, 1997b). Denna ansats rekommenderar bl.a. följande:

- Undvik programmering om möjligt
- Undvik konstanter i kalkylsystemet
- Använd namn i stället för cellreferenser
- Verifiera kalkylsystemet med enkla eller kända värden
- Utnyttja möjligheter till anpassning
- Skydda det som inte skall ändras
- Dokumentera, men lagom
- Utforma ändamålsenliga standarder för organisation och lagring av data (Avdic 1995a:207ff)

En ambition med denna avhandling är att lägga grunden för ett förnyelsesynsätt på utveckling av kalkylsystem. Tumregler av det slag som ovanstående är ett exempel som skulle kunna ingå i såväl traditionalism- som organisationssynsättet, dock enbart som ett komplement till övergripande riktlinjer av annat slag. De riktlinjer som förmedlats ovan kan mycket väl kombineras med mer övergripande riktlinjer vad gäller såväl systemutveckling som IC. Skillnaden mellan tumregler i de olika synsätten är att det enligt förnyelsesynsättet utformas riktlinjer och tumregler utan sidoblickar på eller hänsyn till vare sig systemutvecklingsmetoder eller IC-organisation. I de studerade fallen, se *Del II*, har anvecklingen inte skett vare sig enligt etablerade metoder eller med stöd av någon IC-liknande organisation. Målsättningen att styra anveckling med livscykelliknande metoder eller med IC-liknande organisation har följaktligen inget empiriskt stöd.

10.6 Sammanfattning - Anveckling

Olika forskares uppfattningar om förutsättningar, effekter och lämpligt arbetssätt varierar direkt eller indirekt med synen på styrning och kontroll av anveckling. De faktorer som har inverkan på hur anveckling sprids i organisationer tycks också relaterade till detta. *Spridning* av anveckling kan relateras till kontrollpolicy, förekomst av professionalism (som är förenad med viss självständighet i arbetssituationen), förekomst av informella kommunikationskanaler och utbildning.

Vissa grundläggande *förutsättningar* (i form av verktyg, utrustning, kunskap, uppgifter och viss självständighet i anvecklarens arbetssituation) bör finnas för att anveckling skall vara möjlig. En föränderlig verksamhet med föränderliga informationsbehov berättigar anveckling i en verksamhet. Lämpligt med anveckling är det när anvecklaren har erfarenhet och kunskaper, när verktyget är ändamålsenligt, när anvecklaren har tid, lämpliga arbetsuppgifter, och en självständig arbetssituation. Dessutom skall anvecklaren stödjas av ledning och dataavdelning. God tillgång till välorganiserade data kan också göra anveckling lämplig.

När det gäller anvecklingens *effekter* så ser förespråkare för hög-kontroll organiserandet av anvecklingsarbetet som en väg mot högre systemkvalitet och informationskvalitet. Anvecklaren bör enligt denna syn arbeta efter i princip samma metoder och riktlinjer som systemutvecklare för att utveckla lämpliga system. Enligt hög-kontroll-synsättet kan resursslöseri i form av tid, dubbellagring, suboptimering och direkta kostnader bli en effekt av anveckling om det inte sker organiserat och om det inte sker en styrning av anvecklingsverksamheten.

Enligt en annan syn som förespråkar låg kontroll kan anveckling ses som ett tidsbesparande och ändamålsenligt sätt att tillgodose verksamhetens behov. Låg-kontroll-synsättet poängterar de sociala effekter som anveckling kan ge. Anveckling sker utifrån ett arbetsuppgiftsperspektiv, samt ger ökad individuell och organisatorisk produktivitet. Nya versioner av verktyg, t.ex. kalkylprogram, kan minska behovet av metodik och ökar såväl system- som informationskvalitet. Anvecklaren kan ändra system vid behov. Anveckling kan, enligt låg-kontroll-synsättet, resultera i bättre styrning av verksamheter och ge lokalt inflytande.

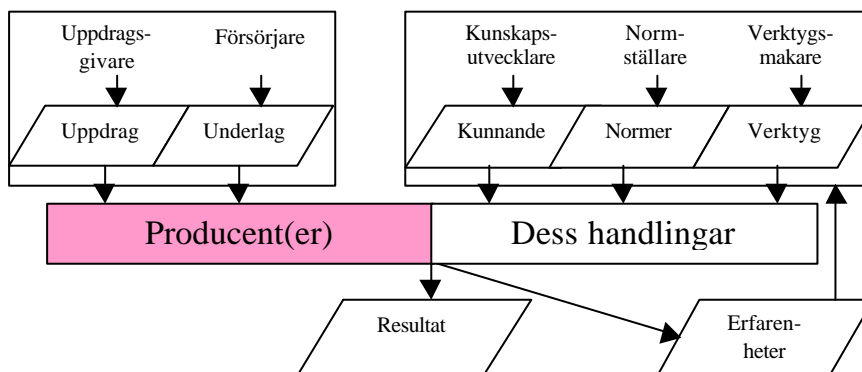
När det gäller inställning till vilket *arbetsätt* som är det mest tillämpliga vid anveckling kan också några synsätt urskiljas. Traditionalismsynsättet ser TSU som en förebild och menar att anveckling bör kunna hämta intryck från denna. Organisationssynsättet ser organisation av anvecklingsverksamheten som lösning på vissa problem. För att finna en benämning på synsätt som inte främst söker samma arbetsätt som de första två har benämningen Förnyelsesynsätt formulerats. Detta synsätt är en paraplybenämning på synsätt som söker finna arbetsätt som främst utgår från andra utgångspunkter än de andra båda synsätten, t.ex. utifrån anveckling som sådan.

Aktörer

"En *anvecklare* är en person som utvecklar informationssystem (a-system) utifrån arbetsuppgifter som ingår i den verksamhet som denne person har god kunskap om och (del)ansvar för. Kunskaperna är både av typen organisationskunskap och av typen professionskunskap."
(Avdic 1999:220)

- Syftet med detta kapitel är att analysera vilka aktörstyper som finns i anslutning till anveckling och kalkylprogramanveckling och vilka egenskaper dessa aktörstyper har.
- Kapitlet syftar även till att analysera och definiera anvecklarbegreppet.

11 Aktörer



Informationsteknikens mest centrala artefakt, datorn, har i likhet med andra artefakter en utveckling bakom sig, där man kan skilja på utvecklarrollen och användarrollen. Denna uppdelning gäller för olika typer av produkter från bilar och båtar, till böcker och teaterpjäser och även för informationssystemutveckling. Mellan rollerna finns en stark koppling. Båtbyggare åker båt och båtåkare kan bygga båtar. En omständighet som gäller för systemutveckling är att hjälpmedel för att utveckla system blivit alltmer lättillgängliga, vilket har inneburit att personer som använder system allt oftare kan utveckla dessa utan hjälp av utvecklare. Inom informationssystemutvecklingsområdet har betydelsen av utvecklare- och användarbegreppen förändrats. Även om nämnda aktörsbegrepp får en annan innebörd i en KPA-miljö än i TSU finns det ändå ett behov att benämna de två kategorierna, ofta kallade *användare* och *utvecklare*. Mer heltäckande och generella uttryck skulle kunna vara *verksamhetsföreträdare* och *IT-specialister*. I avhandlingen används huvudsakligen uttrycket IT-specialist i stället för utvecklare. Ett motiv till att inte använda uttrycken användare och utvecklare som de

mest överordnade begreppen är att de dels ger en något för snäv betydelse och dels associerar till en mer traditionell semantik som inte gäller för anveckling/KPA. En genomgång av olika användning av begreppen ges nedan.

11.1 Aktörer, traditionell systemutveckling

Inom systemutvecklingsområdet finns det olika sätt att benämna och kategorisera de båda grupperna. Sundgren (1992:63) identifierar tre grupper bland verksamhetsföreträdarna (beställarna, slutanvändarna och brukarna). Bland IT-specialisterna identifierar Sundgren fyra grupper (applikationsutvecklare, applikationsansvariga, databasadministratörer och informationssystemsamordnare). Andersen (1994:39ff) talar om användare och utvecklare. När det gäller *utvecklare*, delar Andersen in dessa i systemerare och systemutvecklare. En systemutvecklare sysslar enligt Andersen med "...alla uppgifter inom systemutvecklingen, medan en systemerare enbart arbetar med systemering." Det som Andersen kallar systemering är planeringen av informationssystemet, dvs analys och utformning av informationssystemet, medan systemutveckling utöver detta omfattar realisering och implementering. Användare definierar Andersen som "...experten på det arbetsområde som systemet skall stötta".

I engelsk vokabulär kan uttrycken *systems analyst*, *system operator* och *user* t.ex. Saldarini (1989:645;648), Hicks Jr (1990:81;325) och Beekman (1997:273ff) förekomma. Yourdon (1989:40ff) delar in aktörerna¹ i sju grupper, nämligen:

- Users
- Management
- Auditors, quality assurance people, and 'standards bearers'
- System analysts
- Systems designers
- Programmers
- Operations personnel

Användare (users) kan enligt Yourdon delas in på två sätt, nämligen genom

- Arbetskategori eller ledningsnivå.² Denna nivå innehåller grupperna *operational users*, *supervisory users* samt *executive-level users*.
- Databehandlingserfarenhet.³

Yourdons indelningar utgår från en "traditionell" syn på systemutveckling som projektorienterad och utförd av IT-specialister. Indelningen syftar till att kategorisera aktörer i relation till hur systemutveckling med Yourdons systemutvecklingsmetod skall bedrivas. Yourdons indelning av användare behandlar inte olika roller som användare kan ha i relation till ett informationssystem.

¹ Yourdon kallar dem "players in the systems game".

² "Job category, or level of supervision." (Yourdon 1989:43)

³ "Level of experience with data processing." (Yourdon 1989:43)

Aktörer

Användare kan framstå, om man tolkar det bokstavligt som den som använder programmet eller datorn. Detta kan ge bild av en grupp av aktörer med relativt likartade intressen. I praktiken är normalt så inte fallet, vilket bl.a. Sundgrens delning ovan visar. Goldkuhl (1993:18) specificerar följande aktörer i anslutning till ett informationssystem:

- Handlingsansvariga
- Inanvändare
 - * Ursprungliga formulerare
 - * Förmedlare
- Utanvändare
 - * Förmedlare
 - * Slutliga mottagare
- Verksamhetsklinter
- Övriga berörda

Utöver dessa verksamhetsrelaterade aktörer nämner också Goldkuhl de aktörer som är relaterade till själva datasystemet:

- Dataspecialister⁴
 - * de som utvecklar och vidareutvecklar
 - * de som arbetar med datordriften

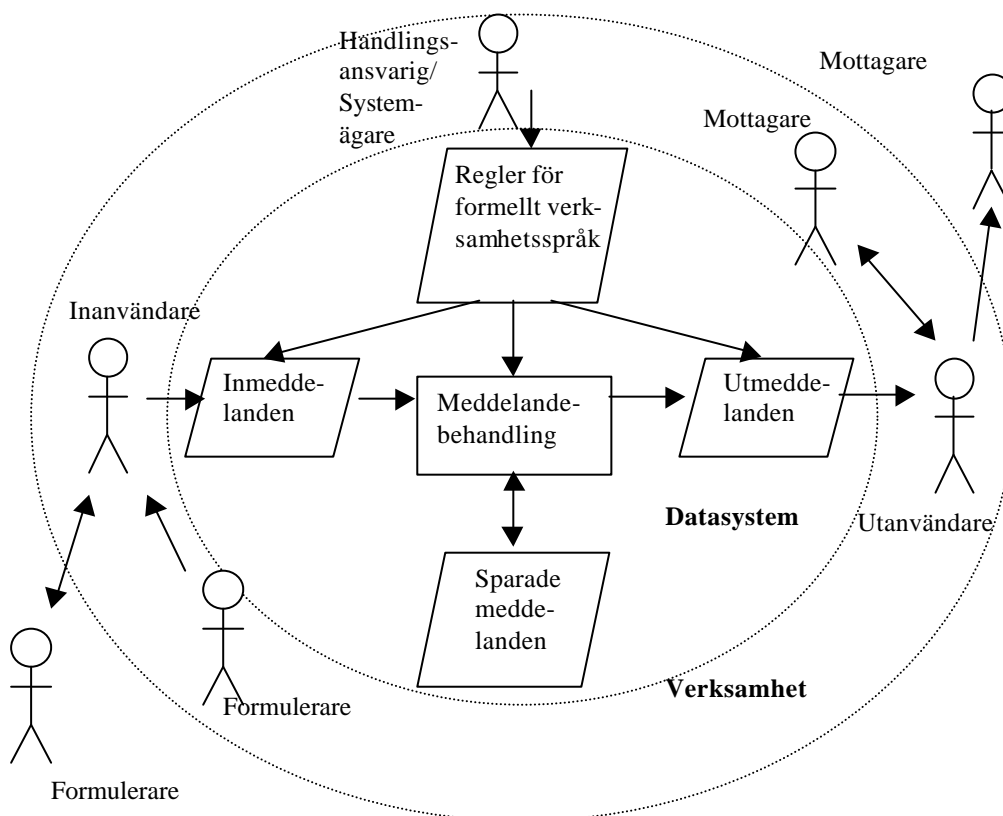
Det som karakteriserar verksamhetsföreträdarna är kunskap om verksamheten i form av kännedom om hur verksamheten bedrivs men också varför den bedrivs som den gör (se kap 1 och 13). IT-specialisternas kunskap omfattar främst kunskap om verktyg för systemutveckling. Verktyg kan i detta sammanhang vara metoder, tekniker, programvara, maskinvara eller annan utrustning. IT-specialister kan även ha god verksamhetskunskap (se t.ex. kap 5.2.5), men deras främsta roll i en organisation är att förse organisationen med specialistkunskap inom IT-området.

Med *handlingsansvariga* avses personer "...som är ansvariga för de funktioner som finns i och utförs av systemet. Man kan också kalla dessa för systemets 'ägare'. Det är de som har makt att formulera och omformulera de regler som datasystemet styrs av."(Goldkuhl 1993:18) Ekonomiska aspekter avseende kostnader för systemutveckling, drift och förvaltning är normalt en angelägenhet för de handlingsansvariga. När det gäller förmedlare (såväl in- som utanvändare) är det de personer som har den direkta kontakten med systemets gränssnitt (oftast tangentbord och bildskärm). Inanvändare-förmedlare behöver inte vara samma personer som formulerar inmeddelanden (ursprungliga formulerare) till systemet och utanvändare-förmedlare behöver inte vara samma person som tar emot utmeddelanden från systemet (slutliga mottagare). Inom TSU är åtskillnaden mellan dessa roller ofta uppenbar. Personer som representerar de olika rolltyperna har olika kunskap ett bidra med i ett utvecklingsprojekt. Beskrivningen ovan visualiseras i Figur 85 nedan.

⁴ I avhandlingen används synonymt uttrycket IT-specialister.

Då ett informationssystem ses ur ett talaktsperspektiv (se sid 161) så utför de olika aktörerna (rollerna) kommunikationshandlingar på olika plan. Ramarna för kommunikationen formuleras av handlingsansvariga i och med att de i systemet inbyggda reglarna fastställs av dessa. Handlingsansvariga bestämmer vad samtalet (via informationssystemet) får handla om. Själva samtalet sker sedan mellan de ursprungliga formulerarna och de slutliga mottagarna.

För förmedlarrollerna gäller att dessa främst berörs av andra aspekter än informationens innehåll i ett informationssystem. Dessa roller berörs av aspekter som berör själva hanteringen av systemet t.ex. transaktionsvolym och systemets användbarhet (se kap 15) i form av t.ex. gränssnittets utformning.



Figur 85 Principbeskrivning av datasystem i verksamhet (Goldkuhl 1993:16).

11.2 Anvecklaren

Den mest centrala rollen i anveckling/KPA är anvecklaren. Benämningen avser att uttrycka en kombination av rollerna användare och utvecklare. Uttrycket *användare* är minst sagt vanligt förekommande i systemutvecklingssammanhang. Nurminen (1987) påpekar att begreppet uttrycker ett konsumentperspektiv i relation till ett informationssystem. En användare använder ett informationssystem och konsumerar information medan andra aktörer (systemutvecklare) tillverkar och underhåller systemet och sålunda producerar informationen. Då människor använder kalkylprogram för att lösa egna arbetsuppgifter kommer de att anta såväl utvecklare- som användarroller. Begreppet

Aktörer

användare blir följaktligen missvisande då man vill benämna den centrala aktören i KPA. Yourdons indelning av användare, enligt a) arbetskategori/ledningsnivå och b) databehandlingserfarenhet, är relevant även för anvecklarbegreppet. Indelning i handläggare, arbetsledare och chef är t.ex. relevant för kalkylsystem som beslutsstöd. Indelningen i databehandlingserfarenhet är relevant för kategorisering av vilka kalkylsystem som kan skapas (se kap 8).

En indelning av anveckling utifrån vem som utvecklar och vem som använder a-systemen visas i Tabell 13 nedan:

Tabell 13 Indelning av anvecklare utifrån vem som utvecklar och använder a-systemet

Typ	Beskrivning
Självförsörjning	En anvecklare utvecklar a-system som endast han själv använder.
Gruppanveckling I	En anvecklare utvecklar a-system som används av en grupp personer där anvecklaren ingår och har verksamhets(del)ansvar.
Gruppanveckling II	Flera anvecklare utvecklar tillsammans a-system som de använder gemensamt där de har verksamhets(del)ansvar.
Fadderanveckling	En (eller flera) anvecklare utvecklar a-system till en grupp personer där anvecklaren har verksamhets(del)ansvar men inte ingår som användare.

Det som är avgörande för anvecklaren är dennes uppfattning om vad verksamheten, som han har ett (del)ansvar för, kräver. Den sista gruppen (fadderanveckling) kan t.ex. vara då en chef skapar ett kalkylsystem som används av en sekreterare. Gemensamt för alla typerna är att utdata kan överföras till andra personer.

11.3 Aktörer, KPA-miljö

I en kalkylsystemmiljö finns som i alla andra mänskliga miljöer ett antal aktörstyper representerade. Eftersom kalkylsystem är informationssystem kan roller i anknytning till informationssystem appliceras på en kalkylsystemmiljö. För att anknyta till Goldkuhls (1993:18) indelning, som är den mest detaljerade av de ovan beskrivna, skulle de kunna indelas på följande sätt:

- **Handlingsansvarig** är den som har ansvaret för den verksamhet som kalkylsystemet fungerar i och som ytterst formulerar eller ansvarar för de regler som kalkylsystemet innehåller. Ansvaret kan finnas på flera nivåer i verksamheten beroende på organisationsstruktur. I en decentraliserad organisation kan handlingsansvar finnas hos enskilda tjänstemän. Ibland kan handlingsansvar vara förenat med budgetansvar. Vid TSU, som beroende på sin projektorienterade natur normalt måste finansieras specifikt, är innehavet av budgetansvar mer centralt än vad gäller anveckling. Vid anveckling är utvecklingsarbetet integrerat i arbetsuppgifterna, varför budgetansvaret inte har samma betydelse. Kalkylsystem kan utvecklas av personer med ansvar för en (del)verksamhet, t.ex. en produktionschef som utvecklar sitt eget produktionsuppföljningssystem (se kap 3.5). Kalkylsystem kan också utvecklas av personer med delegerat handlingsansvar. Detta skulle vara fallet om en handläggare

på ekonomiavdelningen (under ekonomichefen) hade ansvar för budgetprocessen på ett företag och inom ramen för sina uppgifter utvecklade ett kalkylsystem för budgetering. Ett liknande exempel finns i Avdic (1995a:73). Utvecklingen av kalkylsystemet skiljs inte från övriga uppgifter som anvecklaren har, vilket innebär att det ansvar som anvecklaren har gäller för alla arbetsuppgifter inklusive kalkylsystemutveckling. Men då ett kalkylsystem trots allt är ett informationssystem kan man tala om ett systemägarskap. Detta systemägarskap ligger hos anvecklaren.

- **Ursprunglig formulerare** är den som formulerar meddelanden som skall matas in i kalkylsystemet. Exempel på detta är om en kvalitetsansvarig utvecklar ett kalkylsystem för att grafiskt analysera reklamationer och deras orsaker. Om kalkylsystemet innehåller en databas med uppgifter om reklamationer, t.ex. datum, typ, orsak och kostnad så är det den ursprunglige formuleraren, som avgör vilka specifika värden som skall registreras för respektive reklamation.
- **Införmedlare** är den som tekniskt matar in data till kalkylsystemet, t.ex. i form av registrering i en intern databas. Om den kvalitetsansvarige i exemplet ovan inte bara formulerar uppgifter om reklamationer utan även utför registreringen av dem är han en införmedlare.
- **Utförmedlare** är den som tekniskt tar ut data ur kalkylsystemet, t.ex. uttag av diagram eller rapporter. Om den kvalitetsansvarige i exemplet även tekniskt tar fram rapporter och diagram om reklamationer är han en utförmedlare.
- **Slutlig mottagare** är den som erhåller den information som tas ut ur kalkylsystemet. Slutliga mottagare kan finnas i olika led. För ett budgetsystem utvecklat av en ekonomichef kan flera mottagare i olika praktiker identifieras. Ekonomichefen själv kan sägas vara en, verkställande direktör och styrelse en annan, dessutom kan avdelningschefer och handläggare på ekonomiavdelningen vara mottagare liksom revisorer och övriga berörda. För det ovan beskrivna reklamationssystemet kan också ett flertal mottagare i olika praktiker identifieras, produktionsledning, verkställande direktör och styrelse, kunder m.fl. För kalkylsystem kan gälla att mottagare kan finnas både *direkt* och *indirekt*. Om den budgetansvarige i exemplet ovan utformat ett mindre kalkylsystem för att beräkna nyckeltal som skall ingå i budgetsyste-
met, gäller att han själv är mottagare av nyckeltalsinformation, men att mottagare till budgetsyste-
met också är mottagare av nyckeltal, fast indirekt.
- **Utvecklare** är den som bygger och underhåller ett kalkylsystem. Utveckling av eventuell dokumentation hör också till utvecklarrollen.
- **Driftspersonal** är den som ansvarar för drift av nätverk och system. Detta kan omfatta hantering av behörigheter till filer, kataloger och enheter i nätverket. Det kan också gälla backup-hantering
- **Verksamhetsklinter** kallas i dagligt tal, kunder. Eftersom kunden är föremålet för en verksamhet, hela verksamheten riktar sig ju egentligen mot kunden, så är kunden den yttersta användaren av verksamheten inklusive dess system.

Det som kännetecknar anvecklaren är att han har god verksamhetskunskap och viss utvecklingsverktygskunskap. Verksamhetskunskapen som en anvecklare har kan vara av två slag, organisationskunskap eller ämneskunskap. *Organisationskunskap* kan beröra den verksamhet som organisationen bedriver. *Professionskunskap* kan beröra det

Aktörer

yrke som aktören utövar i organisationen. De båda kunskapstyperna kan vara överlappande. Vissa yrken, t.ex. lokförare, finns endast inom vissa typer av verksamheter⁵. Andra yrken t.ex. ekonom, finns inom många verksamheter. Exempel på organisationskunskap skulle kunna vara kunskap om massatillverkning. På ett företag som tillverkar pappersmassa finns många olika yrkesgrupper som alla verkar i en miljö där kunskap om pappersmassatillverkning är väsentlig. Exempel på professionskunskap skulle kunna vara externredovisning. Kunskaper i externredovisning kan utövas på praktiskt taget alla svenska företag.

Utvecklarna, som definitionsmässigt har kunskaper om utvecklingsverktyg, kan vara externa eller interna dels avseende organisationen (dvs ha eller inte ha organisationskunskap), dels avseende en specifik profession (dvs ha eller inte ha professionskunskap). Alla rutor i Tabell 14 nedan, innebär utveckling av kalkylsystem, men det är endast då utvecklaren är intern i båda avseendena som det är fråga om anveckling/KPA. En anvecklare skall följaktligen ha både organisationskunskap och professionskunskap. I tabellen ges exempel på de olika varianterna av kunskapskombinationer. Diskussion om dessa båda kunskapsområden förs i kapitel 1 och i kapitel 13.

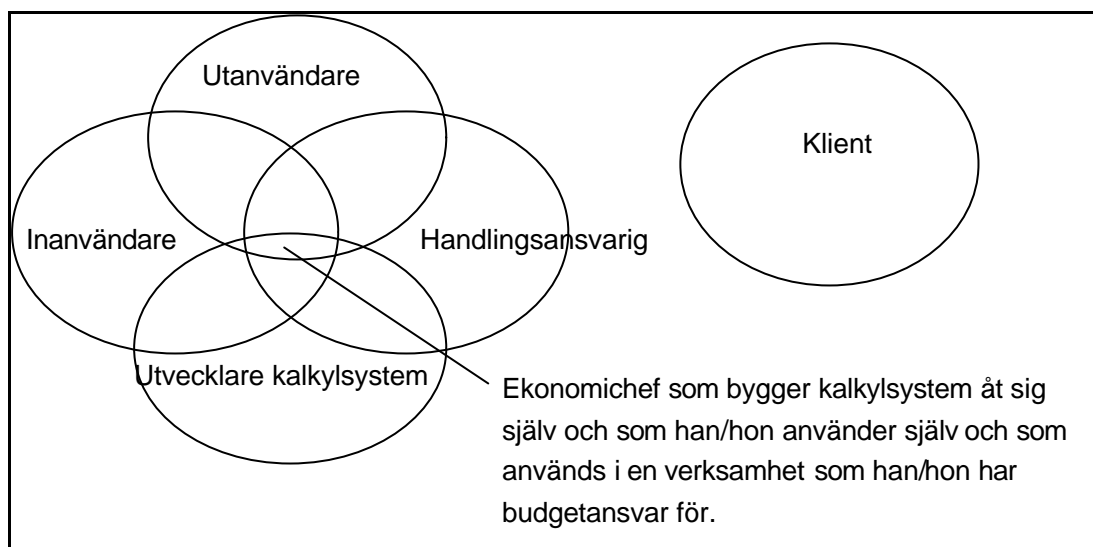
Tabell 14 Olika typer av utvecklare (Avdic 1995a:48).

Utvecklare Interna – externa	Intern avseende Professionen	Extern avseende Professionen
Organisationsintern	En kalkylprogramkunnig ekonomichef som bygger sitt eget budgetsystem	Dataavdelningen bygger budgetsystemet åt ekonomichefen
Organisationsextern	En konsult som är expert på budgetering bygger ett budgetsystem	En IT-konsult bygger ett budgetsystem

Om en ekonomichef bygger ett budgetsystem åt sig själv och som han/hon avser att använda själv, så är han/hon på samma gång handlingsansvarig, inavvändare, utanvändare och utvecklare.

Det typiska med KPA är just att utvecklarrollen kombineras med någon av de tre första (se Figur 86 nedan).

⁵ I och för sig finns lokförare på större industrier med egna industrispår, men tanken med exemplet framgår förhoppningsvis ändå.



Figur 86 Förhållandet mellan aktörer i en KPA-miljö (Avdic 1995a:49).

En annan aktör/intressent som är av speciellt intresse vad gäller standardsystem (ett kalkylprogram är en typ av standardsystem) är leverantören av standardsystemet (Nilsson 1991:8ff). I och med att kalkylprogrammet i sig själv är en del av kalkylsystemet så innebär det att kalkylsystemets funktionalitet avgörs av kalkylprogrammets funktionalitet. Detta innebär i sin tur att lanseringen av en ny version av ett givet kalkylprogram kan påverka möjligheterna att tillgodose informationsbehovet hos en person som bedriver KPA.

11.4 Anvecklare - en definition

En sammanfattande definitionen på begreppet anvecklare lyder:

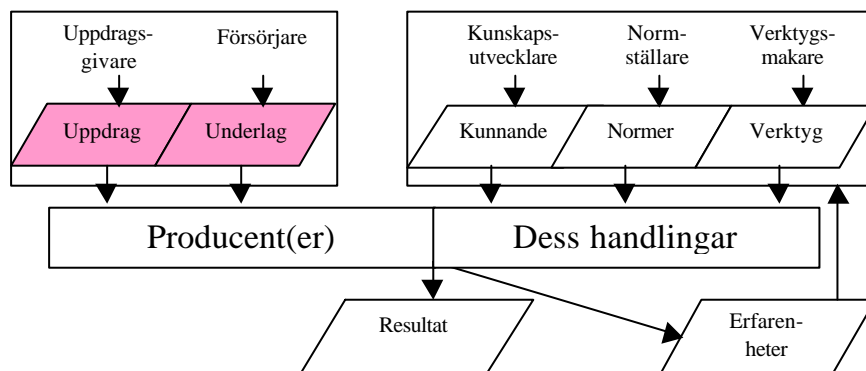
En *anvecklare* är en person som utvecklar informationssystem (a-system) utifrån arbetsuppgifter som ingår i den verksamhet som denne person har god kunskap om och (del)ansvar för. Kunskaperna är både av typen organisationskunskap och av typen professionskunskap.

Att anvecklaren har god verksamhetskunskap och (del)ansvar för verksamheten innebär att han kan hänföras till kategorin professionella. Begreppet professionalism behandlas vidare i kapitel 13.

"Om kalkylprogramanveckling kan förkorta tiden för hantering av beslutsunderlag kan detta följaktligen leda till högre besluts kvalitet."
(Avdic 1999:224)

- I detta kapitel behandlas den praktikgeneriska modellens uppdrags- och underlagsmodul i termer av intention med kalkylprogramanveckling (KPA).

12 Intention och beslutsstöd

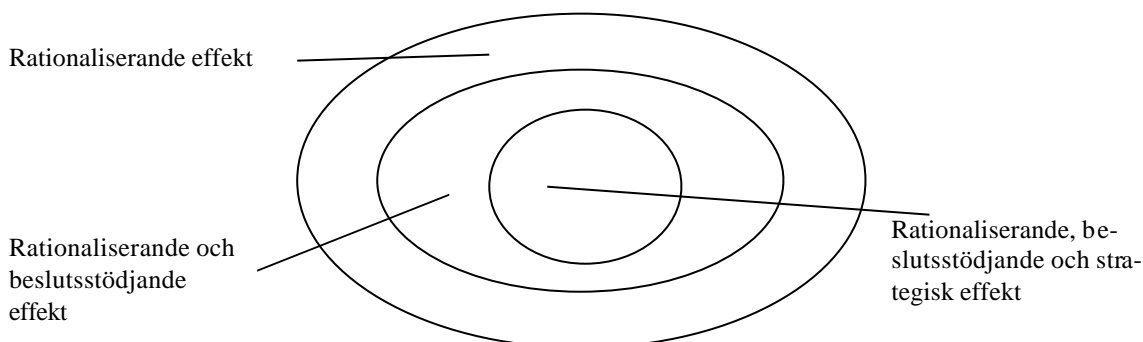


I detta kapitel redogörs för kategoriseringar och teorier inom området beslutsstödsystem, som bedöms ha relevans för (uppdragsgivares) intention med kalkylsystem. Implicit behandlas även underlag. Ett alternativ till att presentera dessa kategoriseringar och teorier under nuvarande kapitelrubrik vore att sprida ut dem i resten av *Del III*. Men då forskningsområdet beslutsstödsystem är förhållandevis väl sammanhållet presenteras det under en och samma rubrik.

Eftersom handlingsperspektivet betonats när det gäller informationssystem och eftersom även integrationen i anveklarens arbetsuppgifter betonats så kan det finnas skäl att reflektera hur man kan betrakta intention med kalkylsystem (se kap 8.3). I Avdic (1995a:189ff) klassificeras *syften med kalkylsystem* som (se Figur 87 nedan)

- Rationaliserande
- Beslutsstödjande-verifierande
- Strategiska

De tre syftena är inte motverkande utan snarare samverkande. Ett kalkylsystem kan uppfylla alla tre syftena. Huruvida systemen uppfyller sina syften visas i form av vilka effekter som erhålls. Effekterna är beroende av varann. I allmänhet har alla kalkylsystem en *rationaliseringseffekt*¹. Automatisering i sig innebär att manuella rutiner ersätts med automatiska, vilket normalt har någon, om än ibland liten, rationaliserande effekt. Rationaliseringseffekter kan t.ex. erhållas när en beräkning automatiseras. Kalkylsystemet producerar då samma utdata som tidigare, men snabbare, säkrare och eventuellt mer tilltalande utseendemässigt. Exempel på detta redovisas i kapitel 4.4, se t.ex. Figur 59 sid 107. *Beslutsstödseffekter* erhålls när utdata ur kalkylsystemet kan skapa, vad berörda aktörer uppfattar som förbättrad besluts kvalitet än tidigare. Ofta är beslutsstödseffekter en följd av rationaliseringseffekter. Om stora mängder för organisationen betydelsefull information kan sammanställas på ett sätt som inte tidigare varit möjligt kan denna effekt uppstå. *Verifierande* effekt är en variant av beslutsstödjande. Med verifierande menas att anvecklaren vill pröva, bekräfta och/eller bygga under åsikter eller teorier som han har. Något nytt tillkommer nödvändigtvis inte, det sker snarast en förstärkning av befintliga ståndpunkter genom att argumenten byggs under. Vikten av noggranna korrekta beslutsunderlag poängterades av samtliga anvecklare i studierna (se t.ex. Figur 26 och Figur 45). Denna aspekt framträder utifrån empirin i *Del II* som en stark drivkraft för anveckling. *Strategisk* effekt kan ett system ha när det påverkar företagets konkurrenssituation på marknaden. Strategi är ett begrepp som förknippas med "...övergripande, generella aktivitetsstrukturer som



Figur 87 Relationer mellan syften med kalkylsystem (Avdic 1995a:191).

sträcker sig över ett längre tidsperspektiv i företag eller organisationer..." (Nationalencyklopedin). I anvecklingsssammanhang innebär en strategisk effekt, en effekt som på verkar företaget på längre sikt på ett genomgripande sätt. Den strategiska effekten kan vara en funktion av antingen en stark rationaliseringseffekt eller en starkt beslutsstödjande effekt eller båda. Att föreställa sig ett strategisk betydelse utan något inslag av de andra två effekterna är svårt. Det strategiska inslaget kan vara av större eller mindre omfattning. Ett exempel på en strategisk effekt i kalkylprogrammiljö är ett byggföretag som automatiserar sin manuella offerthantering till ett kalkylsystem och samtidigt automatiserar inhämtning av prisuppgifter till offertsystemet (Avdic 1995a:132f). Byggföretaget räknar med att minst halvera tiden för offertberäkning, vilket leder till att dubbelt så många offerter kan ges, vilket man räknar med skall ge möjlighet att kunna åta sig fler och eller mer välbetalda uppdrag.

¹ I bemärkelsen effektivisering (Nationalencyklopedin).

Faktorerna som nämnts ovan är inte motverkande. Ett sätt att se på relationerna mellan effekterna är i termer av mängder och delmängder, vilket visas i figuren nedan (storleksproportionerna är inte en följd av systematiska överväganden):

Av de tre ovanstående syftena behandlas här främst det beslutsstödjande. Skälen till detta är två.

Skäl 1: Det faktum att informationssystem kan vara rationaliserande eller strategiska är inget specifikt för kalkylsystem medan beslutsstödssystem och kalkylsystem har en mer nära relation i litteraturen.² Det finns för de övriga två syftena inget motsvarande studieområde som det gör för beslutsstödssystem.³ När det gäller den strategiska effekten, kan den snarast sägas uppstå, då någon av de andra två effekterna nått en viss nivå.

Skäl 2: I referenser som huvudsakligen behandlar beslutsstödssystem diskuteras även andra effekter och syften med informationssystem än beslutsstödjande, vilket innebär att de andra båda syftena behandlas indirekt. Teoribildningen inom området beslutsstödssystem tar t.ex. upp kategoriseringar som: typ av arbetsuppgifter och typ av organisation. Dessa kategoriseringar används för att skilja beslutsstödssystem från andra system t.ex. de som kan uppfattas som enbart rationaliserande.

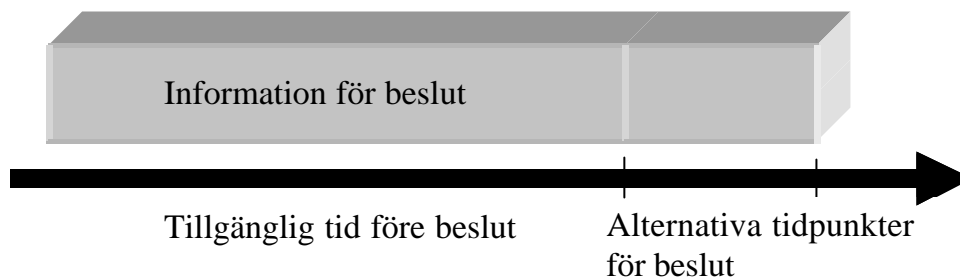
Syftet med ett kalkylsystem är ett uttryck för vilken *effekt* anvecklaren önskar sig i sin verksamhet. Då man talar om syftet med ett kalkylsystem (eller ett informationssystem) så är detta intimt förknippat med de handlingsansvarigas verksamhetsmål och syftet med systemets information (Nurminen 1988:127). Syfte/intention finns hos människor (inte hos informationssystem) men då människors syften (ibland indirekt) manifesteras i informationssystem kan kapitlet relateras till såväl kapitel 8 som till kapitel 11.

En besläktad indelning av informationssystem bygger på en klassificering av den information som systemet behandlar, som operativ eller direktiv (Langefors 1966:155;201, Sundgren 1992:35) där *operativ information* behövs för verksamhetens (omedelbara) drift medan *direktiv information* syftar till att förbättra kvalitet på beslut. Beslutsstödjande system skulle i denna jämförelse kunna sägas behandla direktiv information. Enligt Langefors teori om *exekutiv effektiv information* är det ibland inte mödan värt att sträva efter att beslut skall fattas med all relevant information (Langefors 1967:159). Då beslut måste fattas med information som inte finns omedelbart tillgänglig som underlag, måste detta underlag produceras. En konsekvens är att det tar tid att samla in, kommunicera och behandla information. På grund av t.ex. den tid det tar att samla in all information kan beslutet bli sämre om man tar sig tid att skapa ett komplett underlag än om man avstår från att samla in all relevant information. Innebörden av detta är att beslut ibland kan bli bättre om tiden för att samla relevant information är kortare. Då beslut i allmänhet måste fattas inom en viss tid kan värdet av beslutet rent av minska om beslutsfattaren väntar med att fatta beslutet för att ha

² Flertalet referenser i detta kapitel som behandlar beslutsstödssystem relaterar beslutsstödssystem till kalkylprogram på ett eller annat sätt. I kapitlet ges flera exempel på detta.

³ "Rationaliseringssystem" eller "strategiska system" kan inte sägas vara etablerade benämningar på samma sätt som beslutsstödssystem.

tillräckligt med informationsunderlag. För att utföra uppdraget och fatta beslut med högt värde spelar följaktligen tiden stor roll. En viktig aspekt för förutsättningar för praktik är att de bör stödja insamling, kommunikation och behandling av information för beslut. Sett ur detta perspektiv kan kalkylprogramanveckling ge möjligheter att förkorta tiden för insamlande av underlag för beslutsfattande och utförande av uppdrag, tack vare sin interaktiva och integrerade karaktär. Om KPA kan förkorta tiden för hantering av beslutsunderlag kan detta följaktligen leda till högre besluts kvalitet (se Figur 88 nedan).



Figur 88 Relation mellan tillgänglig tid för beslut och mängden information som kan samlas in, kommuniceras och bearbetas.

12.1 Begreppet beslutsstödssystem

Begreppet *beslutsstödssystem* (DSS)⁴ kan sägas vara en arvtogare till begreppet *Management Information System* (MIS), vilket på 1960-talet representerade en tidig variant av beslutsstödssystem med ambitionen att hantera en hel organisations informationshantering. Förhoppningar fanns att MIS skulle stödja ledningens planerande, operativa och kontrollerande funktioner (Watson m.fl. 1997:10, Nurminen 1988:33ff, 120f). Enligt Watson är DSS, EIS (se nedan) och MIS närbesläktade i sina målsättningar. MIS uppträdde under en tid då de tekniska förutsättningarna inte räckte till för att uppfylla målsättningarna. MIS kom snarast att bli system för att producera standardiserade rapporter.

Beslutsstödssystem definieras av Carlsson (1993:34) enligt:

“Decision Support Systems (DSS) represent a point of view on the role of the computer in the management decisionmaking process. Decision support implies the use of computers to:

- Assist managers in their decision processes in semi-structured tasks.
- Support, rather than replace, managerial judgement.
- Improve the effectiveness of decisionmaking rather than its efficiency.”

I ovanstående definition betonas att beslutsstödssystem stödjer personer som fattar beslut och att arbetsuppgifterna som stöds inte förutsätts vara väl strukturerade. Dessutom poängteras den kvalitativa aspekten av beslutsfattande (effectiveness) snarare än den kvantitativa (efficiency). Flera definitioner poängterar dessa aspekter t.ex. Rey-

⁴ DSS = ”Decision Support System”.

nolds (1992:36), Watson m.fl. (1997:10), Beekman (1997:326) och Capron (1995:296). Andra aspekter som poängteras är modellbyggande, interaktivitet och direkt tillgång till såväl organisationsinterna som organisationsexterna data. En aspekt som är speciell för DSS är att de ofta⁵ förutsätts gälla ledningsfunktioner.

Flertalet av aspekterna kan relateras till KPA, dock ej ledningsperspektivet. Det finns inget i anknytning till KPA som säger att beslutsstödande anveckling måste bedrivas av personer som inte har direkt koppling till lednings- eller chefsfunktioner. Handläggare, tekniker, planerare behöver inte vara chefer för att fatta beslut. Det är snarare beslutsfattandet och ansvaret i relation till detta mer än chefspositionen som gör KPA relevant. Av detta skäl behandlas DSS som att det gäller beslutsfattare i allmänhet och inte endast lednings- och chefspersoner.

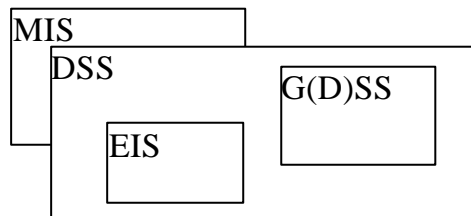
Executive Information Systems (EIS) är en variant av beslutsstödssystem som vänder sig till den högsta organisationsledningen. Ytterligare en variant av beslutsstödssystem är *grupp(besluts)system*. Carlsson (1997) sammanfattar ett antal G(D)SS⁶-definitioner enligt:

Group Support Systems are computer-based information systems used to support intellectual collaborative work

A GDSS is an interactive, computer-based system, which facilitates solution of unstructured problems by a set of decisionmakers together as a group

A GDSS consists of a set of software, hardware, and language components and procedures that support a group of people engaged in decision-related meeting”

Enligt ovanstående definitioner tolkas DSS som ett övergripande begrepp där EIS och GDSS kan betraktas som delmängder av DSS (se Figur 89 nedan).



Figur 89 Relationer mellan MIS, EIS, DSS och G(D)SS.

DSS, EIS och GDSS kan realiserars med hjälp av kalkylprogram. Definitionerna ovan av nämnda typer, täcker väl in flertalet av de kalkylsystem som diskuteras i avhandlingens empiridel, *Del II*.

Stabell (1987:243) ser ledningen (management) i en organisation som DSS-användare. *Management* behöver i detta fall inte nödvändigtvis betyda den högsta organisationsledningen. Om management vidgas till att betyda verksamhetsledning och en verksamhet kan vara en funktion i en organisation t.ex. produktionsplanering, verksamhetsuppföljning eller ekonomiavdelning så vidgas gruppen av användare på ett sätt

⁵ Men inte alltid. Sprague & Watson (1996:6) definierar DSS som: “Computer-based systems that help decisionmakers confront ill-structured problems through direct interaction with data and analysis models.”

⁶ “Group (Decision) Support System”.

som gör att flertalet kalkylprogramanvecklare innefattas, t.ex. de som studerats i kapitel 3 och 4. En något annorlunda syn på användarbegreppet utmärker EIS. E i EIS står för *Executive* som enligt Watson m.fl. (1997:40) har följande kännetecken:

- leder en självständig organisation (en avdelning räknas inte)
- ansvarar för mer än en funktionell enhet
- finns på strategisk planeringsnivå i organisationen
- fastställer policies och representerar organisationen utåt
- har en viktig roll i organisationen
- är inblandade i ett brett fält av angelägenheter inom organisationen.

Att begränsa användningen till executives i den bemärkelse som Watson m.fl. gör innebär att det blir en starkt begränsad skara personer som har att dra nytta av ett EIS. KPA utförs av en vidare mängd personer. Handläggare, ingenjörer, sekreterare, planerare, ekonomer, mellanchefer m.fl. använder kalkylprogram för att införskaffa beslutsunderlag till sig eller andra. Aktörs- och användarbegreppet diskuteras vidare i kapitel 11.

12.2 IT-utveckling ur ett beslutsstödsperspektiv

Kalkylsystem används ofta som en slags beslutsstödsystem (Avdic 1995a). Då DSS och kalkylsystem till stor del är överlappande, kan uppfattningar om utvecklingen av DSS även sägas gälla kalkylsystem och därmed anveckling. Av detta skäl beskrivs nedan IT-utvecklingen ur ett beslutsstödsperspektiv.

Sprague & Watson (1996:7) beskriver DSS som placerat mellan områdena "data processing" och "management science". "Data processing" har utvecklats från enkel databehandling via databaser till frågespråk och rapportgeneratorer. "Management science" har på samma sätt utvecklats från symboliska modeller via datormodeller och modelleringssystem till interaktiv modellering. Detta kan sägas vara en träffande beskrivning även för KPA i och med att kalkylprogrammen fram till dags dato (september 1999) har erhållit mer och mer av den funktionalitet som präglar utvecklingen inom "data processing". Bland utvecklare finns chefer, handläggare, ekonomer, marknadsförare och liknande yrkesgrupper med verksamhet relaterat till "management science". Beslutsfattare kan sägas utföra sitt arbete genom att utföra uppgifter som tidigare utförts av IT-specialister. Ett exempel på detta är Bosses **täckningsbidrag per timme** (se kap 3) som dels innefattar komplicerade beräkningar, vilka vore minst sagt omständliga att genomföra manuellt och som dels hämtar data från centrala databaser.

Enligt Sprague och Watson (1996:7) har följande trender påverkat området DSS.

- Snabbt ökad förekomst av persondatorer inklusive lättanvänd programvara som t.ex. kalkylprogram
- Ökad kapacitet och sjunkande kostnader avseende tele- och datakommunikation
- Ökad tillgänglighet till allmänna databaser och andra externa datakällor
- Snabb tillväxt inom End User Computing (EUC) samt kunskap om IT bland människor i allmänhet

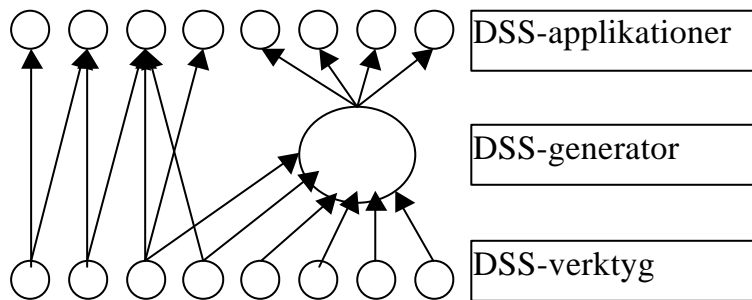
- Ökad tillgänglighet till stora färgskärmar och färg-grafik-programvara
- Ökad tillgänglighet till mobil databehandling och kommunikation
- Tillväxt av AI-tekniker som expertsystem och naturligt-språk-behandling

Nämnda trender har enligt Sprague & Watson fått bl.a. följande effekter på DSS-området.

- PC-baserad DSS har växt, t.ex. har kalkylprogram fått nya analysfunktioner
- Distribuerade DSS har blivit vanligare
- Lokala nätverk och e-post har bidragit till att GDSS växer
- DSS-applikationer har börjat innehålla verktyg och tekniker från AI. Ett DSS kan ”föreslå, lära och förstå”
- En specialisering av DSS har påbörjats. Exempel på detta är EIS och GDSS
- Användarvänligheten hos DSS har ökat

De trender och effekter som Sprague & Watson skisserar, gäller även KPA. I kalkylprogram inkluderas t.ex. funktioner för fleranvändning, åtkomst till interna och externa databaser, funktioner för grafisk presentation, hjälpsystem med inslag av naturligt-språk-behandling, integration med Internet, interaktivt utformade analysverktyg⁷ (se även kap 15.2).

Kalkylprogram kan relateras till andra utvecklingsverktyg och slutanvändarprogram ur ett DSS-perspektiv i termer av *teknologinivåer*. Vid utveckling av DSS kan tre sådana teknologinivåer identifieras. På den högsta nivån finns DSS, dvs den specifika slutanvändarprogramvaran för beslutsstöd. På mellannivån finns DSS-generatorer, med vilka slutanvändaren själv kan skapa ett DSS (anveckling). Exempel på en DSS-generator är kalkylprogram. På den lägsta nivån finns DSS-verktyg, vilka närmast kan jämföras med programspråk. Med DSS-verktyg kan såväl DSS-applikationer som DSS-generatorer utvecklas (Sprague & Watson 1996:14) (se Figur 90 nedan). Ur detta



Figur 90 Tre nivåer av DSS teknologi (Sprague & Watson).

perspektiv kan kalkylprogram (DSS-generatorer) sägas vara utvecklat med hjälp av DSS-verktyg. Dessutom kan DSS-applikationer utvecklas med kalkylprogram (DSS-generatorer). Flera av de kalkylsystem som finns beskrivna i *Del II* (t.ex. Jans **beräkning av kreditivräntor** och Olles **pris- och indexutveckling**) kan enligt detta resonemang klassificeras som DSS-applikationer.

⁷ MS Excel Solver är ett exempel på detta.

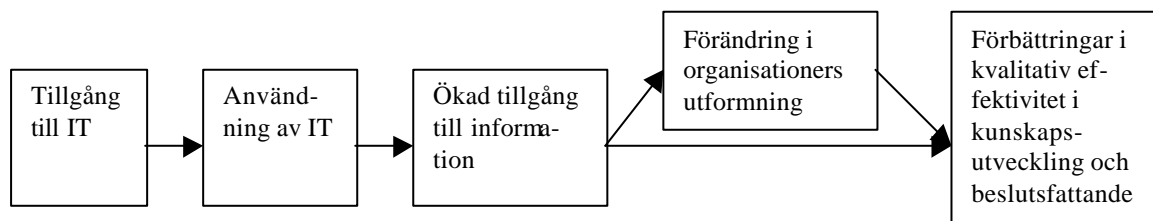
Huber (1990:47-71) har formulerat en teori för hur IT påverkar beslutsfattande. Huber karaktäriserar IT:s natur ("The Nature of Advanced Technologies") som uppdelat i två delar, *grundläggande kännetecken* ("basic characteristics") och *egenskaper* ("properties"). Grundläggande kännetecknen relateras enligt Huber till datalagringskapacitet, överföringskapacitet och bearbetningskapacitet. När det gäller egenskaper menar Huber att dessa, i en *kommunikationskontext*, gör det möjligt för individer och organisationer att

- kommunicera enklare och billigare över tid och rum
- kommunicera snabbare och med högre precision till specifika målgrupper
- lagra och indexera mer tillförlitligt och mindre kostsamt innehåll och typ av kommunikationshändelser
- selektivt kontrollera tillgång till och deltagande i kommunikationshändelser

I en *beslutsstödskontext* ("context of decision aiding") inkluderar egenskaperna möjlighet för individer och organisationer att

- lagra och återfinna stora mängder information snabbt och billigt
- snabbt och selektivt kunna få tillgång till information skapat utanför organisationen
- snabbt och exakt kunna kombinera och omkonfigurera information för att skapa ny information
- använda expertsystem och beslutsmodeller
- lagra och återfinna mer tillförlitligt och mindre kostsamt innehåll och typ av organisationstransaktioner

Sammanfattningsvis hävdar Huber att *tillgång till IT* leder till *användning av IT*. Användning leder till *ökad tillgång till information*. Ökad tillgång till information leder till förändring organisationens utformning. Ökad tillgång till information och förändring i organisationens utformning leder till förbättringar i kvalitativ effektivitet (effectiveness) i kunskapsutveckling (intelligence) och beslutsfattande (se Figur 91 nedan).



Figur 91 Tillgång till IT leder till förbättringar i kvalitativ effektivitet i kunskapsutveckling och beslutsfattande (Huber 1990:70).

Hubers beskrivning av avancerad IT i form av dess grundläggande kännetecken och egenskaper implicerar KPA. När det gäller datalagringskapacitet, överföringskapacitet och bearbetningskapacitet har dessa ökat dramatiskt för kalkylprogram. Detta hänger samman med IT:s utveckling i stort, vilket behandlats i kapitel 1. Vissa av punkterna, speciellt egenskaper i en beslutsstödskontext, kan relateras direkt till vissa former av användning av kalkylprogram, t.ex. att kunna kombinera och omkonfigurera information för att skapa ny information. Den orsakskedja som Huber skisserar leder till, eller syftar till att leda till, förbättringar vad gäller "effectiveness" i organisationen beskriver på ett träffande sätt vad som faktiskt händer när utvecklare utför arbetsuppgifter

och använder kalkylprogram till detta. Samtliga tre anvecklares kalkylprogramanvändning i kapitel 3 kan relateras till Hubers modell. Såväl Bosses **täckningsbidrag per timme** som Thomas **uppföljning av arkningsmaskiner** är konkreta exempel på detta.

12.3 Arbetsuppgifter för beslutsstöd

Beslutsstödssystem förutsätter arbetsuppgifter av beslutsfattande karaktär. Detta skapar ett behov av att precisera hur beslutsfattande arbetsuppgifter är beskaffade. En sådan indelning av arbetsuppgifter görs av Sprague & Watson (1996:7) som skiljer på procedur- och målbaserat arbete. Skillnaderna mellan dessa typer av arbete visas i Tabell 15 nedan.

Tabell 15 Skillnader mellan procedur- och målbaserat arbete (Sprague & Watson 1996:7).

Kategori	Procedurbaserat arbete	Målbaserat arbete
Transaktionsvolym	Hög	Låg
Kostnad/transaktion	Låg	Hög
Struktur på procedurer	God	Dålig
Mått på resultat	Definierade	Mindre väl definierade
Fokus	Process	Problem och mål
Effektivitetsfokus	Efficiency (kvantitativt)	Effectiveness (kvalitativt)
Föremål för hantering	Data	Begrepp
Målgrupp	Administrativ personal (clerical workers)	Ledning och professionella (managers, professionals)

Denna indelning överensstämmer inte nödvändigtvis med en indelning i administrativt arbete ("clerical") och ledning-professionalism ("managerial professional") eftersom administrativ personal mycket väl kan ha målbaserade uppgifter. Enligt denna indelning är det målbaserade arbetsuppgifter som är lämpade för DSS. När det gäller kategorin effektivitetsfokus representerar "effectiveness" en kvalitativ effektivitetsaspekt medan "efficiency" snarast representerar en kvantitativ. Enligt Stabell (1988:13) ingår efficiency i effectiveness. Som tidigare nämnts beskrivs effectiveness som det som eftersträvas via DSS.

De exempel på KPA som redovisas i Avdic (1995b) och i *Del II*, t.ex. Annettes **samanställning av en personalenkät** i kapitel 4.5, kan relateras till ovanstående beskrivning på målbaserat arbete, eftersom typ av bearbetning, val av verktyg och val av presentationsform avgjordes av Annette utifrån ett uppdrag, formulerat på en övergripande nivå.

12.4 Kalkylsystem - DSS

Kalkylsystem är den produkt som produceras vid KPA.⁸ De kalkylsystem som har ett beslutsstödjande syfte omfattas av definitioner av DSS (Watson m.fl. 1997:10, Carlsson 1993:34 m.fl.). Aspekter som betonas bland definitionerna och som kan relateras direkt till kalkylsystem är

- **Dataaccess.** Access till data ska kunna vara god, vilket innebär att man t.ex. skall kunna ha tillgång till interna⁹ och/eller externa databaser. I kalkylprogram finns funktioner för att importera och förädla data från interna databaser. Även möjligheter att nå externa data finns.¹⁰
- **Interaktivitet.** Systemen är interaktiva, vilket innebär att det ställs krav på dialogfunktioner i systemet. Kalkylprogrammet är interaktivt till sin natur. Ett exempel på det är att kalkylbladet räknar om nya indata i samma ögonblick som det matas in.
- **Simulering.** En kvantitativ aspekt, som t.ex. tar sig uttryck genom möjligheter att ställa "what-if"-frågor till systemet. Denna funktion är central i kalkylprogram.
- **Uppgiftstyp.** Används för ostrukturerade icke-rutinartade uppgifter till skillnad från rutinuppgifter som bygger på standardrapporter. Kalkylprogram kan användas till såväl produktion av standardrapporter som till spontana rapporter.
- **Aktör.** Används av beslutsfattare till skillnad från personer som utför standardarbetsuppgifter. I de fall som utvecklaren själv konstruerar kalkylsystemet rör det sig inte om rutinartade uppgifter.

Några aspekter som nämns i andra sammanhang (Sprague & Watson 1996, Stabell 1987, etc.) när det gäller DSS

- **Gränssnitt.** Systemen förutsätts vara användarvänliga, vilket innebär att användare utan specialkunskaper om IT skall kunna använda systemet. Definitionsmässigt är ett kalkylsystem, om inte användarvänligt, så i alla fall användarorienterat, om det är utvecklat av utvecklaren till sig själv.
- **Grafisk presentation.** Det skall finnas möjligheter till grafisk presentation då detta ibland uppfattas som mer åskådligt än enbart siffror och text. Enligt Holmberg (1996:53) är presentation en funktion som med fördel kan utföras med kalkylprogram.

I definitionerna nämns inget om hur systemet utvecklats eller hur stort, komplext eller i vilken miljö det verkar.

12.5 Kalkylprogramaspekter

Kalkylprogram framställs i DSS-litteraturen som ett verktyg för utveckling av DSS (DSS-generator) (Carlsson 1993:60ff, Sprague & Watson 1996:14, Stabell 1987:243). I själva verket är kalkylprogram den enda standardprogramvara som nämns i samband med DSS förutom specialverktyg, speciellt inriktade på utveckling av beslutsstödsy-

⁸ Detta gäller för utvecklaren i sin kalkylsystemutvecklarpraktik, när det gäller utvecklaren i sin verksamhetsföretagpraktik är produkten kalkylsystemets information.

⁹ I bemärkelsen organisationsinterna, t.ex. en databas med produktionsdata.

¹⁰ Ett exempel på detta är MS Excel97 är möjlighet finns att infoga hyperlänkar till Internetsidor, vilket gör det möjligt att kontinuerlig importera externa data, t.ex. börskurser via Internet.

stem (t.ex. Triple C). De funktioner som kalkylsystemet innehåller, enligt diskussionen ovan, måste realiseras med hjälp av kalkylprogrammet. Carlsson (1993:227) menar att det för kalkylprogramanvändning för beslutsstöd ska finnas minst tre uppfyllda villkor. Nämligen

1. en uppgift måste vara central
2. kalkylprogrammet och uppgiften måste matcha varandra
3. den presumtive användaren bör ha ett visst inflytande över arbetsuppgiften och en viss självständighet i sitt arbete

Förutsättningarna som beskrivs ovan visar att kalkylprogram inte bara kan användas som DSS-generatorer utan pekar också på att kalkylprogram kan vara ett relevant DSS-verktyg i vissa fall. Speciellt gäller detta då beslutsfattaren själv använder verktyget. Det är också fullt möjligt att använda kalkylprogram för att i projektform utveckla applikationer till beslutsfattare. Denna användning framstår dock som ovanliga-re.

Angehrn (1991:127f) som har starka band till DSS-verktyget Triple C menar att en ändamålsenlig ansats strävar efter att underlätta beslutsfattarens lärande och att verktyget skall stödja detta. Även Courbon (1995:119) poängterar den lärande aspekten med DSS-utveckling. Enligt den evolutionära prototypingorienterade ansats som Courbon förespråkar, ses varje varv i utvecklingscykeln som en kombination av handling och reflexion, som anses vara de grundläggande elementen i lärande.

Denna syn kan relateras till KPA på så vis att man mycket väl kan se många fall av KPA som just lärande genom handling och reflexion. Då man inom DSS, speciellt DSS-utveckling med hjälp av IT-specialister, som sköter systemkonstruktion, syftar till lärande om verksamheten och beslutsfattandet, gäller lärandet vid KPA även utvecklingsverktyget, dvs kalkylprogrammet. Detta kan beskrivas i termer av samma-n-flätade aktiviteter (Avdic 1995a:181). Anvecklarna i kapitel 3 och 4 är exempel på detta lärande.

12.6 Kommunikation - GDSS

Sprague & Watson (1996:17ff) framhåller att den kommunikativa aspekten får alltmer utrymme i DSS-miljö. Det sker en fragmentering av delområden inom DSS, t.ex. utvecklas electronic meeting rooms mot ett eget kunskapsområde. Man menar vidare att beslutsstödssystem utvecklas mot gruppstödssystem, vilket kan tolkas som att en bild av individer som fattar beslut förskjuts mot en syn på beslutsfattande som ett utbyte av information i grupper av olika slag. Även Courbon (1995:121) framhåller kommunikationsaspekten. "Decision support is becoming communication support". Courbon tror att DSS kommer att utvecklas genom att ta intryck från groupware-funktionalitet.

PC-miljöerna har under 90-talet utvecklats från stand-alone-miljöer till nätverksmiljöer (Sprague & Watson 1996:18;21, Beekman 1997:12f etc.). Nätverken är dels lokala (LAN) och dels mer vidsträckta än så (WAN). I organisationer innebär detta att nya

möjligheter till kommunikation har uppstått, vilka delvis är oberoende av tid och rum. Några exempel på det är:

1. LAN. Ett vanligt lokalt nätverk ger möjligheter att lagra gemensamma data som kan vara tillgänglig för olika användargrupper.
2. Intranät. Ett medvetet utnyttjande av detta lokala nätverk där t.ex. olika avdelningar på företaget informerar de anställda i olika frågor, t.ex. produktionsstatistik eller kontoplaner.
3. E-post. En kommunikationsform som innebär utbyte av meddelanden och dokument i olika kombinationer.
4. Internet och därmed förknippade funktioner. Hämta information från och presentera information på hemsidor, e-post, filöverföring, diskussionsgrupper, chat etc.

Kalkylprogrammets roll i detta kan ses på flera sätt. Rent *tekniskt* så förses kalkylprogram med möjligheter att delta i GDSS. Som exempel kan anföras MS Excel 97 som innehåller stöd för fleranvändning av kalkylblad och möjligheter att skapa html-sidor. En annan aspekt är att kalkylprogram kan *användas av verksamhetsföreträdare*¹¹ i lokala nätverk som gemensamma dokument där användare på egen hand kan förse omgivningen med information. Ett exempel på detta är när inköpschefen kontinuerligt förser marknadsavdelningen och ekonomiavdelningen med aktuell prisinformation via kalkylblad på det lokala nätet (se kap 3.4).

En reflexion i detta sammanhang är att den tekniska utvecklingen har tack vare utvecklingen mot LAN och WAN gjort det möjligt att stödja kollektiva beslutsprocesser och inte bara enskilda beslutsfattare som var den (outtalade men) förhärskande synen innan GDSS blev ett allmänt begrepp. I kapitel 3.4 beskriver Bosse hur det lokala nätverket kan utnyttjas för att verifiera det rimliga i modeller utförda med kalkylsystem, vilket kan ses som ett exempel kollektivt beslutsfattande.

12.7 Sammanfattning - Intention och beslutsstöd

Informationssystem kan ses som formaliserade handlingar och viljeytringar. Detta innebär att informationssystem kan ses som uttryck för mänskliga intentioner (Searle 1969). Området beslutsstödssystem behandlar informationssystem för beslutsstöd. Eftersom kännetecknen på beslutsstödssystem (modellbyggande, interaktivitet, tillgång till data, används för att stödja beslutsprocesser, etc.) passar in på många kalkylsystem (Avdic 1995a:51ff, *Del II*) kan forskningsområdet beslutsstödssystem betraktas som nära relaterat till KPA (Carlsson 1993, Watson m.fl. 1997). Av detta skäl sammanfattas nedan aspekter om beslutsstödssystem som kan anses relevanta för KPA. Dessa aspekter avses att användas i avhandlingens *Del IV* för analys av teori och empiri.

Kalkylprogram kan karaktäriseras som en sk DSS-generator, ett mellanting mellan en applikation och ett verktyg, som gör det möjligt för beslutsfattare att själva bygga beslutsstödssystem (Sprague & Watson 1996).

¹¹ I motsats till IT-specialister.

Intention och beslutsstöd

Enligt Langefors teori om exekutivt effektiv information är det ibland inte mödan värt att sträva efter att beslut skall fattas med all relevant information som underlag. På grund av t.ex. den tid det tar att samla in, kommunicera och behandla all information kan beslutet t.o.m. bli sämre än om man avstår från denna information (Langefors 1966). Denna tidsaspekt gör att förutsättningar som kan förkorta tiden för insamling, kommunikation och behandling ökar möjligheten att fatta bra beslut eftersom man då kan hinna få med ett mer komplett beslutsunderlag på samma tid som tidigare. Alternativt kan beslutet fattas tidigare. Förutsättningar som kan påverka informationshantering för beslut kan enligt den praktikgeneriska modellen vara normer, kunnande och verktyg. Om KPA kan förkorta tiden för hantering av beslutsunderlag kan detta följaktligen leda till högre besluts kvalitet.

Den typ av arbetsuppgifter, som lämpar sig bäst för beslutsstödssystem (och därmed även kalkylprogramanveckling) är målbaserat arbete snarare än procedurbaserat. Målbaserat arbete utmärks av låg transaktionsvolym, hög transaktionskostnad, dålig procedurstruktur, mindre väldefinierade resultatmått, fokusering på mål och problem snarare än process, fokus på kvalitativ effektivitet snarare än kvantitativ, begreppshantering snarare än datahantering, ledning och experter som målgrupp snarare än administrativ personal (Sprague & Watson 1996).

System för kollektivt beslutsfattande kan realiseras med kalkylprogram. Denna utveckling har möjliggjorts av utveckling av PC-miljöer från stand-alone till nätverksmiljöer.

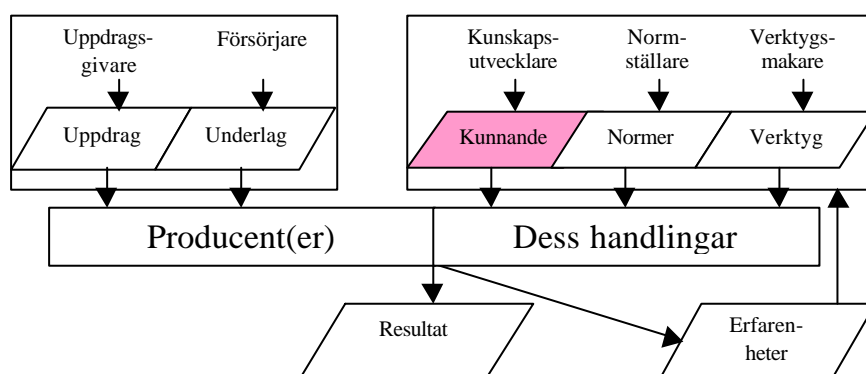
Kunskap

"Anvecklare kan explicitgöra tyst kunskap genom att bygga kalkylsystem, vilket i sin tur gör kunskapen möjlig att värdera och kritisera för en vidare grupp intressenter. I förlängningen kan detta ses som en del i ett kvalitetsutvecklingsarbete i en organisation."

(Avdic 1999:255)

- *Syftet med detta kapitel är att relatera kunskapsbegreppet till anvecklingsbegreppet*

13 Kunskap



I den praktikgeneriska modellen framhålls kunnande¹ som en intern förutsättning för praktik (Goldkuhl 1997:8). I den modifierade modellen som beskrivs i kapitel 7 framställs kunnandet i anveklarens praktik som fördelat mellan anveklaren och verktyget. I avhandlingens inledande kapitel argumenteras för att anveklarens innehav av olika kunskapsstyper spelar roll för hur anveckling bedrivs och kan bedrivas. Två kunskapsstyper, kunskap om utvecklingsverktyg och verksamhetskunskap identifierades. Begreppet kunskap, som är centralt för denna avhandling, både som studieobjekt och som resultat, låter sig inte definieras på ett enkelt sätt. I detta kapitel penetreras kunskapsbegreppet och relateras till studieområdet för att på så vis fördjupa förståelsen för forskningsfrågan och bidra till att besvara den. Syftet är att söka meningsfulla begrepp, strukturer och kategorier med vars hjälp studieområdet kan beskrivas och relateras.

¹ Kunnande definieras som "kunskap och kompetens" (Goldkuhl 1997:3). Betydelsen av kunskapsbegreppet diskuteras i detta kapitel. Kompetens kan här tolkas som "skicklighet, duglighet, kapacitet" (Ord för ord 1964:341). Begreppet kompetens diskuteras vidare på sid 249.

Speciellt väsentligt är kunskap om formalisering av kunskap eller förutsättningar för formalisering av kunskap.

Utgångspunkten för denna avhandling är att människor, som utför arbetsuppgifter i verksamheter, behöver vissa kunskaper för detta. Detta kan enligt Goldkuhl (1997:5) benämnas praktik, vilket som tidigare nämnts definieras ”Någon/några som gör något för någon/några”.

”Centralt i en praktik är *görandet*; olika handlingar. Det kommer att finnas handlingar och som har olika betydelser och relationer till varandra. Görandet är styrt av ändamål; att göra *något*; att producera ett resultat. Av definitionen framgår också praktikers sociala karaktär.” (Goldkuhl 1997:5)

Kunskap ses här i relation till handlandet och till det önskade resultatet. Denna kunskapssyn är pragmatisk. Praktikern har eller tillgodogör sig kunskaper som behövs för att åstadkomma det eftersträvade resultatet. Enligt denna syn är alltså kunskap relaterat till handlande i verksamheter, t.ex. utvecklare som utför arbetsuppgifter. Denna syn illustreras av följande två citat:

”Our knowledge is interwoven with our practice.” (Dahlbom & Mathiassen, 1991:32)

”Knowledge does not belong to a separate cognitive sphere. It is related to practice in various ways.” (Nurminen 1995:214)

Ett delvis annorlunda synsätt är att se kunskap/information som en avbildning av (en del av) verkligheten, snarare än som ett uttryck för intentioner hos intressenter i denna verklighet. Exempel på denna syn kan hittas i Sundgren (1992:79ff) och Bubenko jr & Lindencrona (1984:55ff). Denna ansats upplevs som mindre fruktbar ur anvecklingsperspektivet där fokus ligger på utvecklarnas egna handlingar och intentioner.²

13.1 Kunskapsformer

Molander (1988:142) skiljer på 1) *utförandekunskap* (veta-hur), 2) *påståendekunskap* (veta-att) och 3) *förståelsekunskap* (veta-innebörden-av). Med *utförandekunskap* avses kunskaper om hur vissa typer av *handlingar* utförs, t.ex. klippa en gräsmatta eller skriva ett strukturerat dataprogram. Handlingarna kan även vara gemensamma t.ex. ”...en religiös ceremoni, ett samlag eller ett samtal...” (Molander 1988:143). Med *påståendekunskap* som handlar om hurvida man känner till om ett påstående är sant eller ej, t.ex. att första världskriget började 1939 eller att Excel 97 kan hantera 65 536 rader på ett kalkylblad. *Förståelsekunskap* kan enligt Molander ses som en kombination av veta-att och veta-hur, till exempel att förstå innebörd och användning av ett begrepp. *Förståelsekunskap* finns primärt i gemenskap med andra människor. Även Nordenstam (1993:32) identifierar tre likartade kunskapsformer är i princip synonyma. Med *förståelsekunskap* menas en kunskapsform av ett mera grundläggande slag som i och med sin natur är mer svårbeskriven än de övriga två. Ett skäl till svårigheten att beskriva och fånga in dess natur är att den förutsätter sig själv. För att beskriva *förståelsekun-*

² Därmed inte sagt att de anförda referenserna skulle bortse från eller beakta att aktörer har intentioner. Jämförelsen berör snarast hur frågeställningar fokuseras.

skap krävs förståelsekunskap. *Praktikkunskap*, dvs den kunskap som en producent/anvecklare behöver för att utföra sin praktik består av samtliga tre kunskapstyper.

13.2 Verksamhetskunskap

Att inneha kunskap om en verksamhet framstår som väsentligt när det gäller att agera i denna verksamhet, dvs utföra vissa arbetsuppgifter på ett meningsfullt sätt. Denna verksamhetskunskap är intressant ur det arbetsuppgiftsperspektiv som KPA är nära relaterat till. Verksamhetskunskapen är en typ av praktikkunskap.

I idealfallet agerar människor rationellt i sina arbetsuppgifter i verksamheter mot gemensamma mål och utför handlingar mot bakgrund av sina (1) kunskaper om verksamheten som sådan, (2) kunskaper om verksamhetens mål, (3) kunskaper om hur den specifika arbetsuppgiften kan utföras och (4) kunskaper om regler i samband med utförande av den specifika arbetsuppgiften. Ett exempel skulle kunna vara en ekonom som arbetar i ett producerande företag (t.ex. Bosse, kap 3.4). Ekonomens uppgifter är att följa upp kostnader i företaget för att eventuellt bidra till att minska dessa, där det är förenligt med företagets policy i övrigt. För att kunna utföra uppföljningen bör ekonomen i idealfallet ha (1) vissa kunskaper om produkter och produktion, (2) kunskaper om kvalitetsmål, andra policies, kritiska nyckeltal etc. som är aktuella i verksamheten, (3) kunskaper om hur kostnadsuppföljning bedrivs, beräkning av nyckeltal, fördelning av kostnader etc. (dvs yrkeskunskaper för ekonomer) och (4) kunskaper om regelverk för vad som är lagligt och/eller acceptabelt, t.ex. avskrivningsregler, bokföringsprinciper eller agerande mot andra aktörer i verksamheten. Som diskuterats i kapitel 1 kan verksamhetskunskap delas upp i kunskap som har med verksamhetens specifika (kärn)verksamhet att göra och kunskap som har med anvecklarens profession att göra (se Figur 92 nedan). Områdena kan vara delvis överlappande.

Då ekonomen bedriver sin kostnadsuppföljning, kan han använda kalkylprogram för detta ändamål. Här får ekonomen behov av ytterligare en typ av kunskap, nämligen om användning av datautrustningen inklusive kalkylprogrammet för att åstadkomma de önskade resultaten (se Figur 92 nedan).

Verksamhetskunskap: 1. om verksamheten 2. om verksamhetens mål	Professionskunskap: 3. om professionen 4. om professionens mål	Verktygskunskap: 5. om verktyg
--	--	-----------------------------------

Figur 92 Olika typer av kunskap som en anvecklare behöver.

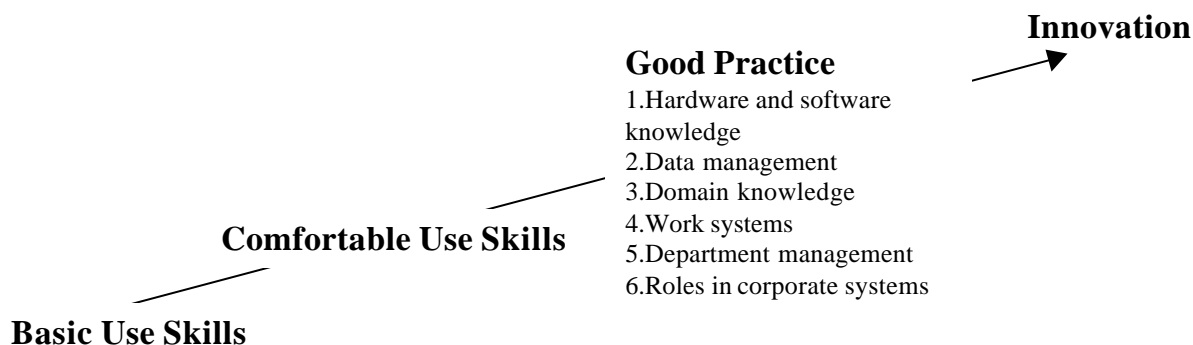
13.3 Verktygskunskap för anvecklare

Panko (1988:167ff) har formulerat en kunskapshierarki³ för anvecklare⁴ (se Figur 93 nedan). Hierarkins lägsta nivå kallas *Basic Use Skills*, som innefattar grundläggande

³ "Skills hierarchy".

⁴ "Users".

kunskap om (1) datorer och operativsystem samt om (2) den programvara som skall användas. Enligt Panko kan denna kunskap erhållas efter två kurser om nämnda området. Den andra nivån kallas *Comfortable Use Skills*, vilket innebär att användaren känner sig hemmastadd med datorn och med den programvara som används. Detta stadium nås enligt Panko genom erfarenhet förvärvad genom användning. På denna nivå kan användaren det som behövs för att utföra en arbetsuppgift, men inte mer. Den tredje nivån kallas *Good Practice*, vilket innebär att användaren har tillägnat sig mer kunskap än vad som minimalt behövs för att kunna utföra sina arbetsuppgifter. Nivån inbegriper kunskaper som behövs för att kunna använda datorn och programmen effektivt och säkert. Exempel på sådana kunskaper är datamodellering, katalog- och filorganisation, programutvecklingskunskaper. Här behövs också verksamhetsrelaterad kunskap som domänkunskap⁵, kunskap om arbetssystem⁶, kunskap om ansvarsförhållanden⁷ och kunskaper i korporativa datasystem⁸. Den fjärde och sista nivån kallas *Innovation*, vilket innebär att utvecklaren kan förändra organisationens sätt att arbeta utifrån kunskaper om den nya teknologin. Detta kan sägas vara en variant av *IT som möjliggörare*.



Figur 93 Taxonomi för utvecklarkunskaper (Panko 1988:167).

Relaterat till indelningen i olika typer av utvecklarkunskap (verksamhetskunskap, professionskunskap och verktygkunskap) som visas i Figur 92 ovan, så kan de lägsta två nivåerna (basic och comfortable use skills) relateras till verktygkunskap. På nivå tre (good practice) kan 1 och 2 relateras till verktygkunskap, 3 (domänkunskap) kan relateras till professionskunskap och 4, 5 och 6 till verksamhetskunskap. Den högsta nivån, Innovation, kan relateras till samtliga kunskapstyper.

En annan kunskapshierarki, som endast berör verktygkunskap och som utgår från att utvecklaren har djup verksamhetskunskap, kommer till uttryck genom den kunskap som behövs för att konstruera informationssystem av olika komplexitetsgrad. Avdic (1995a:121) har identifierat kunskapsbehov som behövs för olika kalkylsystemtyper. För den enklaste typen, *Ordbehandlersystemet* krävs att användarutvecklaren kan de allra mest grundläggande funktionerna i kalkylprogrammet som att starta och avsluta,

⁵ "Domain knowledge" innebär kunskap om specifik arbetsuppgift eller professionsrelaterad kunskap.

⁶ "Work systems" innebär ungefär befintliga arbetsuppgifters organisation.

⁷ "Department managers", berör ansvar, beslutsordningar och policies.

⁸ "Roles in corporate systems" berör hanteringen av större datasystem som går över avdelningsgränserna, t.ex. ekonomisystem.

Kunskap

spara, skriva in text, enkel redigering, skriva ut dokument (för exempel, se Avdic 1995a:123ff). Däremot krävs t.ex. inte att han skall förstå hur en katalogstruktur är uppbyggd i och med att man inte behöver organisera data på sekundärminne för att kunna skapa denna typ av kalkylsystem. För att bygga *Lilla kalkylsystemet* krävs förkunskaper motsvarande de som gäller för *Ordbehandlarsystemet* och dessutom förmåga att skriva enklare formler och använda funktioner, i synnerhet funktionen SUMMA. Ett exempel på ett litet kalkylsystem är **beräkning av arkantal per pall** (se sid 67). Att kunna göra enklare⁹ diagram kan också räknas till förkunskaperna för *Lilla kalkylsystemet*. För att bygga *Stora kalkylsystemet* krävs ovan beskrivna förkunskaper samt förmåga att behärska mer avancerade deklarativa funktioner som t.ex. att länka kalkylblad, skriva komplexa formler, skapa avancerade diagram¹⁰ etc. Exempel på stort kalkylsystem är **produktkalkyl** (se sid 77). För denna typ av kalkylsystem ställs också större krav på förståelse av datorns arbetssätt och hur data kan lagras på sekundärminne. För att bygga ett kalkylsystem av typen *Applikationen* krävs ovan beskrivna förkunskaper samt förmåga att utnyttja det inbyggda behandlingsspråket, dvs programmeringskunskaper. Exempel på applikation finns i kapitel 6.

I en studie av Westerlund (1997), som syftar till att beskriva relationer mellan kalkylprogramkunskaper och "...möjligheterna att skapa lyckade kalkylsystem" (Westerlund (1997:7) diskuteras effekter av anvecklarens verktygskunskapsnivå. Westerlund formulerar resultatet i termer av ändamålsenlighet, enkelhet och flexibilitet. Med *ändamålsenlighet* avses "Samma jobb som förut men snabbare. Flera tidigare uppgifter i samma system. Ger tydliga underlag. Tillförlitlig information." (Westerlund 1997:11). Med *enkelhet* avses "Överblickbar in- och utdata. Liknande arbetssätt som tidigare. Kräver inga större justeringar i yrkessituationen. Självständighet i systemet." (Westerlund 1997:11). Med *flexibilitet* avses "Möjlighet att (smidigt) anpassa till nya omständigheter. Påverkas inte (i stor utsträckning) av att detaljer läggs till eller tas bort. Möjligt att anpassa för nya krav på systemets produkt." (Westerlund 1997:11). Följande effekter noterades för anvecklare med låg respektive hög verktygskunskap (se Tabell 16 nedan). Westerlunds studie behandlas vidare i kapitel 17.1.

⁹ Med "enkla" avses diagram som utgår från ett på kalkylbladet sammanhängande område, och vars serier innehåller värden från samma intervall.

¹⁰ Ett avancerat diagram skulle kunna vara ett diagram som innehåller olika serietyper, ett som hämtar data från olika områden på ett eller flera kalkylblad eller ett som innehåller avancerad formatering t.ex. staplar med bilder.

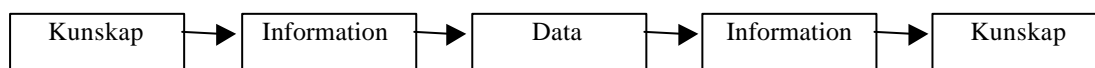
Tabell 16 Effekter på kalkylsystem beroende av anvecklarens kunskaper om kalkylprogram (Efter Westerlund 1997:33).

Egenskap	Effekt om anvecklare har <u>låg</u> kunskap om kalkylprogram	Effekt om anvecklare har <u>hög</u> kunskap om kalkylprogram
Ändamåls-enlighet	Kalkylsystemen kännetecknas av att beräkningar som utförs av systemet är slutprodukter, ingen vidare förädling eller analys sker i systemet.	Kalkylsystemen kännetecknas av att de initiala beräkningarna är ett första led i informationsutvinningen, därefter finns möjlighet att vidare förädla informationen eller utföra olika analyser av den.
Enkelhet	Högre tid för systemförvaltning	Lägre tid för systemförvaltning
Flexibilitet	Lägre flexibilitet	Ökad systemflexibilitet i och med att anvecklaren kan identifiera och definiera variabler och minimera antalet konstanter vid formelkonstruktion.

13.4 Kunskap, information och data

Det är människor som har kunskap. Kunskap kan förmedlas som information. Eller som Dahlbom & Mathiassen (1991:30) uttrycker det, "Information is something you request and receive, knowledge and competence is something you have." Data används för att representera information, t.ex. med bokstäver. Information och kunskap är begrepp som är förknippade med människor. Information är tolkade data och måste därför tolkas av en människa för att bli till information. Information kan bli till kunskap hos människor. Någon har kunskap och vill förmedla detta till någon annan varvid denne får (delar av) den avsedda kunskapen (se Figur 94 nedan). Många fallgropar finns dock på vägen mellan att en människas kunskap blir till en annan människas förvärvande av denna kunskap. Som diskuterats i kapitel 1 (Langefors infologiska ekvation) är en risk vid informationsöverföring att informationen inte tolkas av mottagaren såsom sändaren avsett. Mottagaren kan sakna vilja eller förmåga att tolka informationen som sändaren avsett. Orsaker till detta kan vara olika referensram eller skillnader i språk.

"Turning knowledge into information and information into data can be a difficult task as soon as we aim beyond anything but the most formalized and routinized type of knowledge. Sometimes it is quite impossible." (Dahlbom & Mathiassen, 1991:36)



Figur 94 Förhållande mellan kunskap information och data i ett kommunikationsperspektiv.

Data är uttryck för en semantisk koppling till begrepp som i sin tur refererar till fenomen i verkligheten¹¹. Dessa semantiska relationer åskådliggörs med hjälp av Ogdens triangel i Figur 95 nedan. Den streckade linjen markerar en indirekt referens mellan en

¹¹ Förutsatt att man omfattar föreställning att det finns en verklighet, vilket förutsätts här.

Kunskap

term och dess motsvarighet i "verkligheten" (via begrepp). Linjen mellan termer och begrepp representerar den tolkningsaspekt som gör att data blir till information och till någon form av kunskap hos den mottagande parten i en kommunikationshandling.



Figur 95 Ogdens triangel (Ogden & Richards 1949)¹²

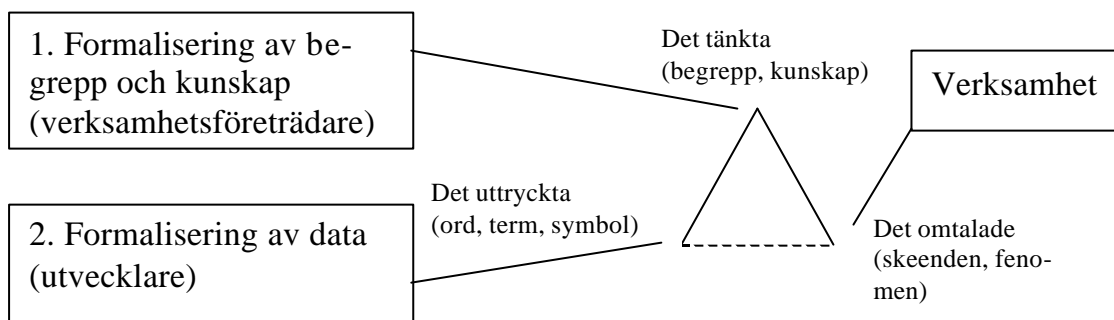
Omvandlingar mellan kunskap, information och data är centrala aktiviteter inom IT-området. Att låta information och kunskap erhålla olika representationsformer är en förutsättning för insamling, lagring, bearbetning, överföring och presentation med hjälp av datorer. En anvecklare som har kunskaper om sin verksamhet (det omtalade) kan utnyttja kalkylprogram för att hantera (delar av) denna kunskap på olika sätt. Anvecklaren formaliserar sin kunskap på olika sätt för att kunna uppfylla målen med sina arbetsuppgifter, där det t.ex. kan ingå att informera någon intressent om verksamheten i något avseende. Med formalisering menas här att anvecklaren specificerar informationen på ett entydigt sätt genom att formulera entydiga regler. Ett exempel på detta kan vara om en anvecklare har som uppgift att bevaka valutakursutveckling för att minimera lånekostnaden på ett företag (Avdic 1995a:67). Om anvecklaren använder kalkylprogram för att lagra uppgifter om lån och kurser måste uppgifterna formaliseras så att de kan lagras. Om anvecklaren vill utforma ett kalkylsystem som förutom att lagra uppgifter också skall bidra till utförande av analyser och simuleringar, så måste de önskade beräkningarna anta formen av formler, funktioner eller program av något slag. De regler som anvecklaren anser vara ändamålsenliga måste explicitgöras och formuleras entydigt för att vara meningsfulla.

13.5 Svårformaliserbar kunskap

Att formalisera kunskap och information är ett centralt moment inom systemutveckling. Formalisering förutsätter att kunskap/information görs explicit (Dahlbom & Mathiassen, 1991:36). Formaliseringen av kunskap/information är nödvändig för att den skall kunna registreras, lagras, behandlas, överföras och presenteras. Formaliseringen fyller även i TSU syftet att möjliggöra kommunikation om innehållet i kunskap/information. Om t.ex. ett reklamationsuppföljningssystem är under utveckling i en verksamhet och flera intressenter är inblandade, kan det vara lämpligt att de ingående begreppen formaliseras utifrån ett verksamhetsperspektiv. I reklamationssystemet kan man t.ex. vilja införa olika felkoder för olika feltyper för att kunna fördela kostnader på feltyper. Om syftet med systemet är att minska kostnader och öka kvaliteten i produktionen så kan man förvänta att formaliseringen sker utifrån dessa mål. Detta

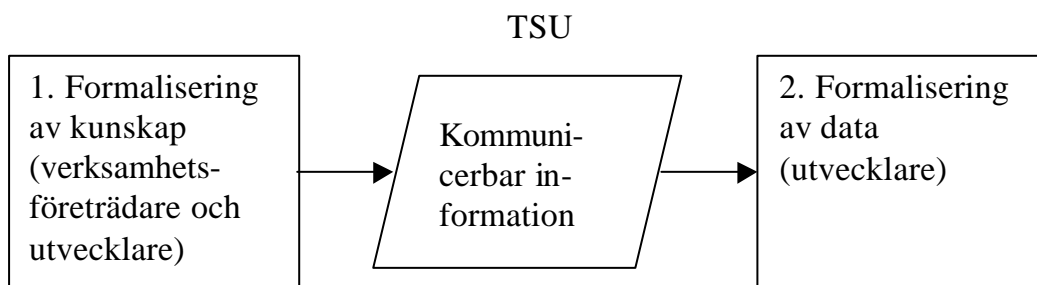
¹² Referensen är indirekt. Ogdens triangel är omtalad i åtskilliga böcker och artiklar. Den indirekta referensen är Goldkuhl, Nilsson & Röstlinger (1982:10).

representerar en pragmatisk kunskapssyn. Formalisering av informationens innehåll (begrepp, kunskap) är en första förutsättning för formalisering av informationens representation (dvs data) (se Figur 96 nedan). Den första typen av formalisering (1) sker med hjälp av aktörer med verksamhetskunskap, kunskap om verksamheten och dess mål krävs för att kunna abstrahera relevant information. Den andra typen av formalisering (2) kräver kunskap om användning av utvecklingsverktyg och tillhör utvecklarens uppgifter. När det gäller TSU tillhör formalisering (2) IT-specialisternas arbetsuppgifter. Även formalisering (1) räknas som en uppgift för IT-specialister i TSU, dock med nödvändig hjälp av verksamhetsföreträdare. Vid anveckling kan det räcka med den andra typen av formalisering. Ett av kännetecknen på en anvecklare är ju att han redan innehar verksamhetskunskaper. Anvecklaren behöver utifrån sin utvecklarroll inte formalisera verksamhetskunskaper på samma sätt som systemutvecklaren när han utvecklar. I anvecklarens primära yrkesroll ingår att utföra de arbetsuppgifter som är föremål för anveckling, vilket förutsätter goda verksamhetskunskaper. I praktiken kan även systemutvecklaren ha goda verksamhetskunskaper (se kap 5.2.5).



Figur 96 Formaliseringsuppgifter för (1) verksamhetsföreträdare och (2) utvecklare utifrån Ogdens triangel.

Om en utvecklare utför formalisering (1) tillsammans med en verksamhetsföreträdare måste den formaliserade informationen göras *kommunicerbar* innan formalisering (2) sker.¹³ I själva verket är utformning av underlag för informationssystemutformning en central del vid TSU. För att kunna utföra formalisering (2) måste informationen vara möjlig att analysera och diskutera (vara kommunicerbar) (se Figur 97 nedan).



Figur 97 vid TSU måste underlag för systemutformning vara kommunicerbart.

¹³ T.ex. i form av en kravspecifikation.

Kunskap

Då en anvecklare utvecklar ett (kalkyl)system kan de båda formaliseringsaktiviteterna utföras enligt anveklarens preferenser. Anveklaren kan själv bestämma om han önskar specificera underlag för formalisering av data (utformning av informationssystem) eller inte. Formalisering kan ske växelvis (integrerat) (se Figur 98 nedan).

Anveckling

Formalisering av begrepp och kunskap och formalisering av data (verksamhetsföreträdare och utvecklare = anveklare)

Figur 98 Vid anveckling kan formaliseringsaktiviteter ske utan att viss information, t.ex. underlag i form av verksamhetskunskap, görs kommunicerbar.

Kunskapsformer kan ur ett informationssystemutvecklingsperspektiv, ses som mer eller mindre formaliserbara (och kommunicerbara). Kunskap kan ändå utnyttjas i en systemutvecklingsprocess om den innehåser av den som utvecklar systemet. Påståendekunskap är en kunskapsform som traditionellt formaliserats och behandlats i form av data(baser) i informationssystem (Nurminen 1988:128). Exempel på formaliserad påståendekunskap är ett kundregister. Utförandekunskap kan vara formaliserbar i form av programvara (och data). Exempel på formaliserad utförandekunskap är Problemlösaren i MS Excel som optimerar numeriska modeller utifrån en algoritm som bygger på linjär programmering. Kunskapsformen förståelsekunskap är mer svårformaliserad. Förståelsekunskap bygger på mänskliga egenskaper som inte enkelt låter sig specificeras. Forskningsområdet AI behandlar formalisering av mindre lättformaliserade kunskapsformer. Exempel på detta är expertsystem som strävar efter att formalisera tumregler för vissa typer av beslut inom t.ex. det medicinska området.

13.6 Tyst kunskap

En typ av svårformaliserbar kunskap är tyst kunskap. Molander (1996:35) skriver angående *tyst kunskap*: "...vi vet och kan mer än vi kan uttrycka i ord." Bland forskare finns olika uppfattningar huruvida begreppet tyst kunskap är relevant. "Man kan nämligen argumentera för att ingen kunskap är helt tyst och för att all kunskap är, i grunden, tyst." (Molander 1996:40) Utifrån den indelning i kunskapsformer som beskrivs ovan (påståendekunskap, utförandekunskap och förståelsekunskap) kan påståendekunskap sägas vara minst tyst. Själva benämningen (påståendekunskap) antyder att det rör sig om kunskap som manifesteras genom påståenden. Det kan dock hävdas att påståendekunskap inte entydigt uttrycker företeelser eller fenomen. "Ingen kunskap ligger i satserna i sig. Det finns något att veta endast om man *förstår* de ingående begreppen och de *användningssammanhang* som satserna normalt hör hemma i, vilket är något annat än att kunna upprepa dem som vilken papegoja som helst." (Molander 1996:40) Utförandekunskap kan överföras mellan människor genom exempel och härledning i en mästare-gesällrelation och behöver följaktligen inte (kan inte fullt ut) formuleras explicit. Förståelsekunskap är komplex till sin natur och därmed inte enkel eller möjlig att formulera eller formalisera. Utifrån diskussionen om kunskapsöverföringsproble-

met (se kap 1) är det/de fenomen, som tyst kunskap fokuserar, relevant(a) för denna avhandling. En bärande tanke bakom avhandlingen är att svårformaliserbar (tyst) kunskap kan utnyttjas bättre vid anveckling än vid TSU.¹⁴

13.6.1 En inledande innebördsdiskussion

Tyst kunskap kan beskrivas som icke-verbal och personlig, och som uttryck för ”...*element i forskningens värld bortom det intersubjektivt prövbara.*” (Mårtensson & Nilstun 1988:48) Michael Polanyi uppmärksammade ovan nämnda svårformaliserbar a kunskap i sin bok *Personal knowledge* (1958). ”*Personlig kunskap uppstår när kunskapstraditioner förenas med individuell upplevelse. Samspelet mellan individ och tradition resulterar i det Polanyi kallar ‘personlig kunskap’*” skriver Rolf (1991:13). I sin bok *Profession, tradition och tyst kunskap* analyserar Rolf (1991) begreppet personlig kunskap med Polanyis kunskapsuppfattning som utgångspunkt. Analysen är väl relaterad till denna avhandlings studieområde eftersom kunskapsbegreppet och överföring av kunskap är centralt.

Uppmärksammandet av personlig kunskap och därmed förknippade fenomen och problem står i viss kontrast till en rationalistisk-analytisk vetenskapstradition. Anledningen är att de delvis positivistiskt influerade kriterierna kvantitativitet, objektivitet och prövbarhet inte kan anses tillämplbara på begreppet personlig kunskap, på grund av svårigheten att artikulera denna kunskap. En viktig delmängd av personlig kunskap är tyst kunskap.

13.6.2 Invändningar mot begreppet tyst kunskap

Tyst kunskap är inte verbalt artikulera. Om den vore det skulle den inte vara tyst. Hur formulerar man kunskap, som behövs för att bygga en fiol eller en båt på ett entydigt sätt? I och med att tyst kunskap inte är formulerad är det inte enkelt att argumentera kring det. Detta är också en invändning mot begreppet tyst kunskap. Kan man säga att tyst kunskap verkligen finns, om den inte uttrycks eller rent av inte går att uttrycka och inte heller går att argumentera kring? En annan invändning mot tyst kunskap är att kunskap är verbaliserad kunskap och att tänkande bygger på verbalisering av kunskap. Denna syn innebär att tänkande är omöjligt utan någon form av språkligt symbolsystem. Rolf kallar denna syn för *radikal konventionalism* (Rolf 1991:31).

Kunskap kan vara tyst av olika skäl. Den kunskap som är tyst därför att den som har kunskapen inte vill dela med sig eller är inkompetent att dela med sig, räknas här inte som tyst kunskap. Skillnaden kan dock vara svår att avgöra. Även om det förefaller uppenbart att, människor innehar kunskaper av olika slag, som inte utan vidare låter sig artikuleras, kan det vara svårt att bedöma karaktären på och relevansen av den aktuella kunskapen. Den tysta kunskapen kan bli ett försvar för ofreflekterade eller tveksamma handlingar och åsikter. Rolf (1991:53) menar att det kan vara svårt att skilja tyst kunskap från tyst dumhet. Å andra sidan kan inte dessa frågetecken vara ett skäl

¹⁴ När så är möjligt av andra skäl, t.ex. systemets omfattning eller komplexitet.

att bortse från att tyst kunskap kan vara ett väsentligt inslag i många professioners karaktär och särart. Om man slår fast att tyst svårartikulerbar kunskap är ett sådant väsentligt inslag, samtidigt som användning av datorer i yrkeslivet ökar, så uppstår frågan hur denna tysta kunskap skall inverka på användning av datorer.

13.6.3 Olika aspekter av tyst kunskap

Rolf beskriver utifrån Polanyis arbete fyra aspekter på tyst kunskap:

”*Den fenomenella aspekten.* Tyst kunskap är kunskap som inte befinner sig i fokus. Den fenomenella aspekten har att göra med att tyst kunskap riktar vår tanke och varseblivning.”

”*Den funktionella aspekten.* Tyst kunskap är det kringvetande som erfordras för att vi ska kunna ha annan kunskap. Det är den som är förknippad med redskapsaspekten.”

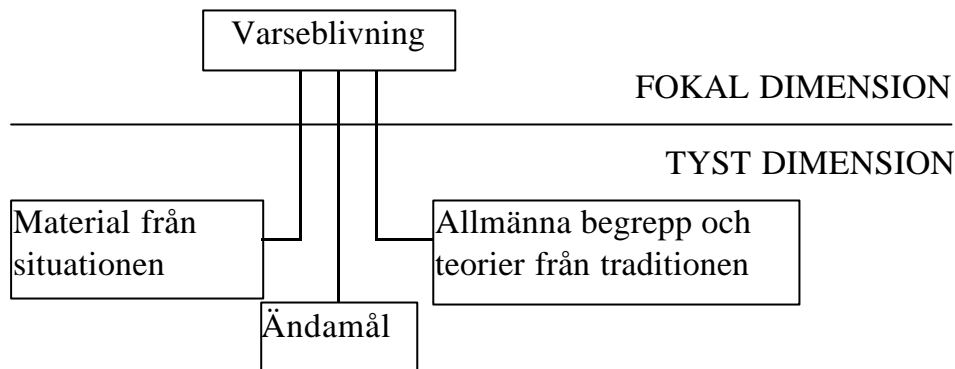
”*Den semantiska aspekten* enligt vilken tyst kunskap ger oss ledtrådar till meningen hos en helhet. Vår kunskap syftar på en meningsfull, begreppslagd enhet.”

”*Den ontologiska aspekten.* Tyst kunskap om delarna bygger upp kunskap om helheten.” (Rolf 1991:64)

Sett ur ett anvecklarperspektiv är *fenomenell* tyst kunskap sådan kunskap, som gör att anvecklaren fokuserar på specifika fenomen och företeelser. En produktionsplanerare i en komplex verksamhet (t.ex. Ismo, se kap 3), kan t.ex. ha kunskaper om olika former av prestanda i produktionsapparaten, olika led, som tillsammans styr handlandet och som t.ex. varnar för situationer som kan leda till oönskade effekter. Ur den *funktionella* aspekten är kunskaper om produktionsapparaten, insikter om produktens egenheter, kundernas förväntningar, vädrets inverkan, olika aktörers förväntade agerande mm, redskap som möjliggör produktionsplanering. Ur den *semantiska* aspekten kan tyst kunskap av nämnda slag göra produktionsplanerarens tillvaro begriplig och meningsfull. Att t.ex. förstå innebörden i begreppen mängdverkningsgrad och tidsverkningsgrad samt innebörden i dessas relation till varandra kan vara en följd av tyst kunskap. Ur den *ontologiska* aspekten byggs produktionsplanerarens kompetens upp av de olika typerna av tyst och icke-tyst kunskap. Att denna typ av dimensioner existerar blir ytterst påtagligt för en person utan verksamhetskunskap, då diskussion förs om verksamhetsrelaterade fenomen. Då Ismo och jag diskuterat produktionsplanering eller uppföljning av denna, kan Ismo plötsligt avfärda ett förslag från min sida med en kommentar av typen ”Det går inte begriper Du väl!” En jämförelse kan göras mellan två personer, där den ene behärskar ett språk och den andre håller på att lära sig språket. Den som kan språket kan förstå ironi och anspelningar på historia, kultur och samhällsfenomen, trots att detta är implicit och kanske aldrig har yttrats verbalt, medan den andre är helt oförstående. Ovanstående aspekter på tyst kunskap och deras relation till KPA utvecklas vidare på sid 309.

13.6.4 Fokal och tyst dimension

Rolf (1991) beskriver (med Polanyi som grund) tyst kunskap som ett fundament, på vilket all kunskap vilar (se Figur 99 nedan). Den fokala dimensionen står i motsats till den tysta. I den fokala dimensionen är varseblivning möjlig.



Figur 99 Tyst kunskap som fundament (Rolf 1991:67).

Sett ur detta perspektiv framstår i princip alla aktiviteter som problematiska att artikulera till fullo, vilket överensstämmer med Langefors' infologiska ekvation (se sid 9).

Sett ur produktionsplanerarens perspektiv (se kap 3.3) kommer i den fokala dimensionen impulser från marknadsavdelningen om vad man lovat olika kunder att leverera till olika datum. Från produktionsavdelningen kommer information om reparationer, kassationer, skiftplanering, produktionsstatus, mm. Från företagsledningen kommer budget och kvalitetsplaner som skall gälla för året eller annan tidsperiod. Från andra källor kommer ett flöde av information om olika fenomen som berör planeringen i större eller mindre utsträckning. Utifrån sin tolkning av alla dessa impulser, utifrån sin uppfattning om sin roll, utifrån kunskap om tillgänglig situation handlar produktionsplaneraren i sin yrkesroll. Den komplexa kombination av utgångspunkter och prioriteringar, som ligger till grund för de produktionsplaner som skapas är till stor del omöjlig att artikulera och därmed formalisera.

Utifrån ovanstående diskussion kan olika reflexioner göras. Produktionsplanerarens arbete blir å ena sidan svårt att värdera och kritisera om det bygger på oartikulerade grunder eftersom det kan vara svårt att hitta konkreta punkter att argumentera kring. Det blir den kritiserandes ansvar att rekonstruera de grundläggande delarna i arbetsuppgifterna. Å andra sidan kan det vara lätt att kritisera verksamhet som inte bygger på artikulerade grunder, eftersom detta faktum i sig kan kritiseras.¹⁵ En annan reflexion är att produktionsplanerarens tysta kunskap, om den är relevant, gör det t.ex. möjligt att följa upp rätt variabler för att kvalitetsgranska resultatet av produktionen. Om produktionsplaneraren skapar kalkylsystem för analys av produktion utifrån sina verksamhetskunskaper kan detta innebära ett explicitgörande av det som tidigare varit tyst kunskap. Detta explicitgörande innebär att den tidigare tysta kunskapen till viss del blir kommunikerbar och kan leda till fördjupning av förståelsen för verksamheten hos en vidare grupp intressenter (se t.ex. Figur 28). Med Rolfs begrepp så flyttas delar av

¹⁵ En annan sak är hur kritiken är beskaffad.

produktionsplanerarens tysta dimension till den fokala dimensionen, som är gemensam för flera intressenter. Produktionsplaneraren i kapitel 3 strävar just efter ett sådant explicitgörande av vad han uppfattar som osäkra erfarenhetsregler.

13.6.5 Objektivisering av det tysta

Personlig kunskap förenar tradition och subjektivitet inom en person. Individens subjektivitet, som visar sig bl.a. via samvetet, balanseras av traditionens inverkan. Traditionen bygger på en gemensam verklighet, vilket innebär att personlig kunskap inte enbart är en personlig angelägenhet. Personlig kunskap är så integrerad att den uppfattas som självklar och har därmed fått en tyst funktion, som människor använder för att handla och ordna sina erfarenheter (Rolf 1991:96f). Ett annat perspektiv på samma fenomen presenteras av Berger & Luckman (1979), som beskriver hur det mänskliga vardagslivet får mening genom att delar av det objektiveras i olika nivåer. Den lägsta nivån benämns habitualisering där det rör sig om individers vanor. Nästa nivå, institutionalisering innebär att flera människor delar uppfattning om i verkligheten förekommande, av människor utformade fenomen (artefakter). Den mest långtgående formen av objektivisering är reifiering, där fenomenen objektiverats så långt att det inte längre är synligt, att det är fråga om artefakter, dvs mänskligt skapade produkter. Berger & Luckman benämner nämnda uppfattningar som uttryck för "...kunskap som styr uppträdandet i vardagslivet" (1979:31). Det finns paralleller till tyst kunskap genom att det rör sig om kunskap som är problematisk att artikulera alternativt tas för given. Den process som styr hur vi inordnar vardagslivet i mönster för att göra det begripligt kan sägas ha ett element av tyst kunskap i och med att den inte alltid kan göras explicit. Varseblivningen i den fokala dimensionen i Figur 99 ovan kan, i viss utsträckning, ses som en objektiviseringsprocess enligt Berger & Luckman (1979:48ff) och där den tysta dimensionen styr vår varseblivning.

Kalkylprogramanvecklaren har arbetsuppgifter att utföra. En del av dessa kan utföras med hjälp av kalkylprogram. I den fokala dimensionen sker varseblivning som styrs av tyst kunskap. Verksamheten kan ses som en objektiviseringsprocess där arbete med kalkylprogram kan bidra till att objektivera aspekter av arbetsuppgifter. Då dessa aspekter objektiveras i formen av kalkylsystem/informationssystem sker objektivisering på en högre nivå, eftersom dessa aspekter formaliseras och permanentas (i olika utsträckning). "De mänskliga uttrycken för känslor och tankar kan objektiveras, det vill säga att de manifesterar sig i produkter av mänsklig aktivitet som är tillgängliga både för dem som producerar dem och för andra människor som element i en gemensam värld." (Berger & Luckman 1979:48) Objektivisering kan sägas vara ett sätt att föra kunskap från den tysta till den fokala dimensionen (se Figur 100 nedan).



Figur 100 Objektivisering, överföring av kunskap från den tysta till den fokala dimensionen.

Thomas kalkylsystem **uppföljning av arkmaskiner** (se kap 3.5.4) kan ses som exempel på detta, i och med att det kan ses som ett sätt att utforska och explicitgöra kunskap, som man vet finns där men som inte är känd eller inte objektiverad.

13.7 Praktisk kunskap

Teoretisk kunskap kan i jämförelse med praktisk ses som föränderlig och osäker, medan praktisk kunskap kan ses som definitiv och påtaglig.

”Görandet är *ett*, oåterkalleligt, definitivt. Man kan spekulera om olika möjligheter och tolkningar. Handlingen är ändå den punkt då vi lämnar de blotta möjligheternas rum och gör på *ett* sätt – världen ändras, det finns ingen väg tillbaka.” (Molander 1996:17)

I en tillverkande industri (t.ex. industrin i kap 3) blir den praktiska kunskapens påtaglighet mycket påtaglig. När produkten väl är producerad finns ingen väg tillbaka. Kunskaperna som användes för att planera och tillverka kan förändras för framtida produktion men den utförda produktionen kan inte göras ogjord. I varje givet ögonblick är handlingar en funktion av kännedom om mål i form av hänsyn till kunders önskemål, kvalitet, lager, etc. samt kunskaper om förhållanden och företeelser, t.ex. produktionskapacitet, transportvägar, bemanningsmöjligheter etc.

Praktisk kunskap är utförandekunskap¹⁶, som ”...vilar på regler med en tyst funktion.” (Rolf 1991:98) Ett exempel på detta är en cyklist som inte vet vilka regler som gör att han kan cykla. Ett annat kan vara en produktionsplanerare som planerar en komplex produktion. Praktisk kunskap är sålunda en del av den personliga kunskapen. Att utöva färdigheter innebär att utöva aktiviteter. Dessa aktiviteter förutsätter mål, som är självklara (och ibland tysta) delar av den verksamhet i vilken de utförs. ”Man kan inte samtidigt utöva verksamheten och aktivt ifrågasätta dess grundläggande värderingar.” (Rolf 1991:98) Med andra ord: ”Medvetandet är alltid intentionellt; det har alltid en intention eller inriktning på objekt.” (Berger & Luckman 1979:32) Praktisk kunskap är sålunda målstyrd även om utövaren inte alltid gjort dessa mål explicita. Enligt Rolf kan praktisk kunskap uttryckas som förmåga till goda handlingar, vilket innefattar tre villkor:

- Förmåga till handling
- Förekomst av vissa kvalitetskriterier
- Dessa kriterier utövas i handlandet

”Praktisk kunskap manifesteras inte genom tur” (Rolf 1991:113)

Att integrera kunskap är även det en sorts färdighet. Att kunna reflektera över dessa färdigheter och dess mål och göra dem explicita, innebär att kunskapen kan analyseras och förmedlas och därmed delas med andra. Tyst kunskap som artikuleras blir reflekterad kunskap. ”Människans kulturellt förmedlade kunskap växer väsentligen, menar Polanyi, genom artikulation av tyst kunskap och reflektion över den.” (Rolf 1991:100) Färdigheten att kunna artikulera tyst kunskap, som även den kräver tyst kunskap, kal-

¹⁶ Kan även kallas färdighetskunskap.

Kunskap

las av Rolf (1991:102) för *andra ordningens tysta kunskap*. Förvärvande av praktisk kunskap är en färdighet som inte går att avyttra, den följer kunskapsbäraren. Även om den praktiska kunskapen inte går att avyttra, så kan andra förvärva den. Ett sätt att överföra praktisk kunskap och de regler som hör till är genom *exempel*. Lärlingen lär av mästaren genom att studera och delta i rutiner och visioner. Här fungerar traditionen som ett kunskapsförmedlande system (Rolf 1991:127ff). "En tradition förutsätter ett underliggande regelsystem, som legitimerar innehållet och definierar sociala roller eller karaktärer. ...Till traditionen hör endast den som definieras som traditionstillhörig enligt det sociala kontraktet." (Rolf 1991:161f) IT-området har en kort och dynamisk tradition som förefaller att ständigt omprövas. Ett IT-exempel på automatisering av kunskapsöverföring genom exempel (och tradition) är program för självstudier. Dessa kurser förmedlar kunskap om program (eller annat), genom sinnrika program som kan fastställa kunskapsnivå på eleven utifrån en formaliserad uppfattning om vad man skall förväntas kunna av en programvara. Möjlighet till test, återkoppling och uppmuntran (!) finns också inbyggt i programmen. Denna typ av programvaror formaliserar såväl kunskaper om programvaruanvändning som pedagogik i viss utsträckning. Ett annat exempel är den typ av on-line-hjälp som finns i t.ex. kalkylprogram. Anvecklaren kan använda denna hjälp för att utveckla kalkylsystem för sin verksamhet. I och med att anvecklarens arbetssituation är självständig kommer själva lärandet att bli föremål för intresse för honom. Samtliga anvecklare i kapitel 3 och 4 var fokuserade på att förvärva kunskaper om kalkylprogram och andra medel, som skulle hjälpa dem att utveckla system i sin verksamhet. I själva verket bestod intervjuerna till viss del av diskussioner om hur olika arbetsuppgifter skulle kunna lösas med kalkylprogram eller andra medel. Anvecklarna strävade efter att förvärva en form av ovan nämnda andra ordningens tysta kunskap i syfte att lära sig hur kalkylprogram skall kunna användas för att utföra arbetsuppgifter.

13.8 Skill, know-how och kompetens

Enligt Rolf (och Polanyi) kan praktisk kunskap, enligt en annan indelningsgrund, delas in i tre slag, vilket redovisas i Tabell 17 nedan (se även sid 260).

Tabell 17 Indelning av praktisk kunskap (Rolf 1991:114).

Typ av praktisk kunskap	Definition
Skill/skicklighet	Skicklighet i utövandet av sådana regler vars efterlevnad jaget kan kontrollera självt.
Know-how utan reflektion	Följande av regler där det sociala kollektivet bidrar med reglerna.
Kompetens = know-how + återkopplande reflektion	Den enskilde kan ge återkoppling till den socialt burna kunskapen genom att ändra dess regler. Det professionella kollektivet kan ändra dess regler.

Exempel på *skill/skicklighet* är färdighetskunskapen cykling. Cyklisten märker själv när han faller. Exempel på *know-how* är förmåga att skämta. En skämtare behöver (normalt) bekräftelse från sin omgivning för att skämtet skall anses som framgångsrikt.

Exempel på *kompetens* är en civilekonom som påverkar bokföringsregler i organisationen han arbetar i och/eller civilekonomförbundets policy eller annan praxis.¹⁷ När det gäller kompetens talar Argyris & Schön (1978:29) om "en andra ordningens inläring"¹⁸, vilket innebär en återkoppling där den kompetente genom sina aktiviteter påverkar det regelsystem som finns. Kompetens kan också i vissa fall fungera som broms mot förändring av områdets regler (Schön 1982:19). Det kan finnas olika skäl för enskilda och/eller grupper att behålla ett regelverk inom ett kompetensområde. Detta diskuteras i termer av verksamhetsmotverkande chauvinism i kapitel 1.2.5.

Exempel från IT-området på de tre kunskapstyperna skulle kunna vara (1) en person som gör sin egen hushållsbudget med kalkylprogram, (2) en byggnadsingenjör som utformar en kalkylsystem för offertkalkyl, som blir efterfrågat i branschen och (3) en systemutvecklare som utformar en systemutvecklingsmetod med egenskaper som gör att den börjar användas av andra systemutvecklare.

13.9 Professioner

Enligt Rolfs definition finns kompetens inom en profession¹⁹. I samhället har professionella grupper olika funktioner. Olika grupper baseras på olika regler och värden och en grupp kan därför inte överta en annans funktion. "Inom professionen förekommer ömsesidig social koordination som ligger till grund för kvalitetsregler och praktisk kunskap." (Rolf 1991:163) Kompetensen i en profession kan sägas vara relaterad till en kunskapsdomän, som är en arbetsuppgiftsrelaterad konstruktion av kunskap av olika karaktär (praktisk, teoretisk, mekanisk etc.). Den gemensamma relationen till denna kunskapsdomän kan utgöra en "professionell identitet" (Molander 1996:13). En profession skall enligt Schön (1982:23) ha fyra huvudegenskaper. Den skall vara specialiserad, välavgränsad, vetenskaplig och standardiserad. Ju mer standardiserad en profession är desto större grund finns för enhetliga kvalitetsregler. Advokater och läkare kan ses som exempel på standardiserade kompetenser.

Den professionella kompetensen bygger på en typ av insikter som är svåra att förmedla. Denna kompetens kan manifesteras då den professionella kunskapen tillämpas på nya förutsättningar. Om den professionella kunskapen är svår att förmedla är den svårformaliserad. Med svårformaliserad menas här inte svårformaliserad i största allmänhet oberoende av hänsyn till vilket sammanhang, utan svårformaliserad i den bemärkelsen att kunskapen skall kunna formaliseras i ett verktyg (t.ex. ett kalkylsystem) för att kunna utnyttjas i ett annat specifikt sammanhang. Här gäller olika nivåer. För verktygsmakaren som skapar och vidareutvecklar kalkylprogrammet gäller operativsystemet och programspråskompilatorn som verktyg. Dessa verktyg, (operativsystemet och kompilatorn) innehåller normer och kunskaper av ytterst allmängiltig karaktär. Det

¹⁷ Eller genom att utforma datoriserade bokföringsprogram av typ Hogia.

¹⁸ Rolfs översättning av "double-loop learning".

¹⁹ "...i allmänt språkbruk detsamma som yrke; i samhällsvetenskaplig terminologi ofta snävare definierat som yrke vars auktoritet och status bygger på hög formell utbildning, ofta universitetsbaserad." (Nationalencyklopedin) Här används en vidare definition. Alla yrken som kan tänkas ha kompetens betraktas som professioner. Exempel på sådana professioner skulle kunna vara produktionsplanerare, hantverkare, ingenjörer, ekonomer eller sekreterare. Se även nedan.

Kunskap

verktyg som verktygsmakaren på detta plan skapar (kalkylprogrammet) är av allmän karaktär och innehåller generiska kunskaper och normer, tillämpbara på en lång rad verksamheter. När det sedan gäller verktyget i nästa nivå, kalkylsystemet formaliseras kunskaper till en mer avgränsad verksamhet. Kunskaperna som formaliseras här kan gälla t.ex. produktionsplanering och vara specifikt avpassade till detta (se Tabell 18 nedan).

Tabell 18 Abstraktionsgrad för IT-verktyg.

Nivå	Abstraktionsgrad	IT-motsvarighet
1	Hög	Operativsystem, kompilator
2	Mellan	Kalkylprogram
3	Låg	Kalkylsystem

Dessutom kan det antas att dessa människor kan integrera skill och know-how från IT-området som verktyg för att utföra sina arbetsuppgifter. Annorlunda uttryckt så tar en profession del av en annan.

Anvecklare finns inom yrkesgrupper av olika slag. Ekonomer, tekniker av olika slag, administratörer och handläggare kan vara anvecklare. En anvecklare måste verka inom en profession där det finns tillräckligt med självständighet att kunna välja om han vill använda ett verktyg av typen kalkylprogram för att lösa arbetsuppgifter. Anveckling gäller professioner eller professionsliknande arbetsuppgifter. Oprecisa arbetsuppgifter med oklart artikulerade mål kräver flexibla verktyg. I de fall som uppgifterna innehåller informationshantering, krävs informationshanterande verktyg. Anvecklaren kan då utveckla tyst kunskap om kalkylprogram för att utveckla sin tysta kunskap om sin verksamhet. Den tysta verksamhetskunskapen kan även göras till reflekterad via den tysta (eller artikulerade) IT-kunskapen.

Hur integrationen av färdigheter från IT-området kan gå till är en fråga som är relevant för denna avhandling. Den kan delas upp i följande två frågor om kopplingar mellan anveckling och tyst kunskap.

- 1) Hur kan personer som tillhör en profession ta del av (relevanta delar av) IT-professionens kunskap?
- 2) Hur kan anveckling utnyttjas för verksamhetsprofessioner för att utveckla/ta hänsyn till/formulera/etc. tyst kunskap?

13.10 Praktik, kunskap och handling

”Det är utmärkande för kunskap i handling, kunskap *i* handling, att kunskapsbildningen samtidigt är ett skapande och upprätthållande av handlingsidentitet – och därmed individens identitet som handlande och varande. Det är inte en kunskap om något skilt från kunskapsbildningsprocessen.” (Molander 1996:241)

Inledningsvis i kapitlet hävdades att kunskapssynen i denna avhandling innebär att kunskap betraktas i relation till handlande och till resultat. Detta innebär i sin tur att praktikbegreppet får en central roll i avhandlingen eftersom praktik innefattar intention-

nella handlingar. Enligt Nordenstam kan praktik ses som "...any kind of rule-following behavior." (Nordenstam 1985:6) Nordenstam skiljer mellan *stängda* och *öppna* regler, där stängda regler är sådana som kan utföras av maskiner medan öppna regler inte kan det. Exempel på stängda regler är matematiska beräkningar av olika slag. Öppna regler är sådana som t.ex. avgör semantiken i ord. För att kunna utöva praktik med öppna regler krävs förtrogenhet med begrepp, något som kräver ett mänskligt medvetande.

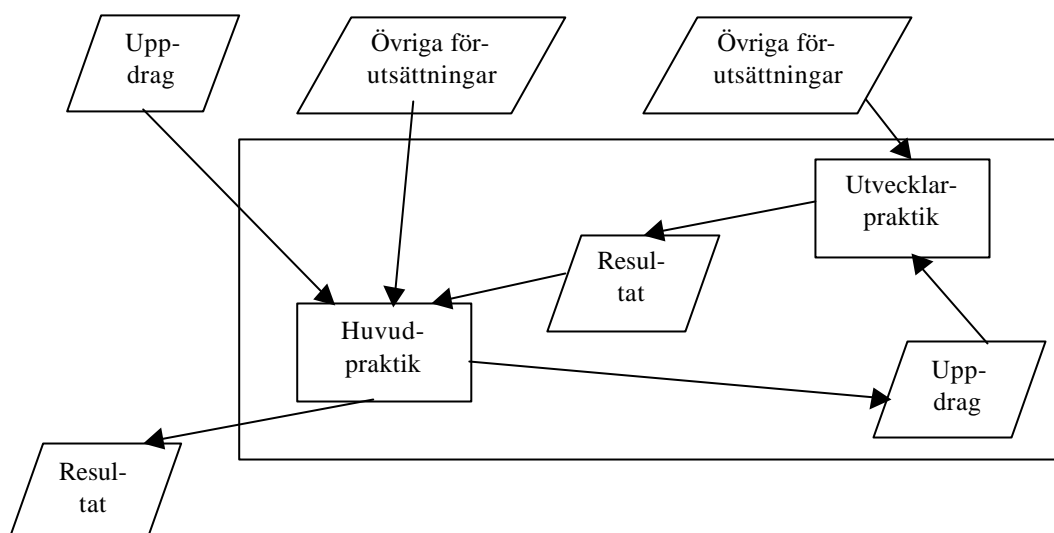
Definitionsmässigt hanterar TIS stängda regler, medan olika ansatser har aktualiserats för att hantera öppna regler med hjälp av informationssystem (se nedan). En anvecklares praktik kan omfatta såväl stängda som öppna regler, beroende på vilken praktik det rör sig om. Ju mer öppna regler desto svårare är praktiken att formalisera. Tyst kunskap vilar i stor utsträckning på öppna regler.

Informationssystem kan ses som informationsskapande och kommunikativa handlingar, där aktörer utför olika typer av handlingar i form av rutiner i informationssystemet. Informationssystem kan ses som uttryck för formella verksamhetspråk (Goldkuhl 1993:156). Ur denna synvinkel innebär systemutveckling utveckling och formalisering av verksamhetspråk i form av informationssystem. De kommunikativa handlingarna, som är formaliserade i informationssystemet, kan ha olika intentioner. Goldkuhl (1995:71ff) ger exempel på en indelning av handlingstyper i en kommunikativ handlingsmodell av affärsrelationer och affärsprocesser.²⁰ Exempel på handlingstyper är erbjudande, fråga, order, bekräftelse, kontrakt, åtagande, fullföljande. Genom identifiering av aktörer och intentioner i form av handlingstyper för meddelanden i informationssystem, fördjupas förståelsen för såväl informationssystemets roll som för systemutvecklingsprocessen.

För a-system och därmed kalkylsystem får den kommunikativa aspekten olika betydelse beroende på huruvida anvecklarens praktikresultat (se Figur 101 nedan) är underlag i en praktik med samma fysiska person eller ej. I det fall en anvecklare utformar ett verktyg för sin egen verksamhet överväger i det första ledet den informationsskapande aspekten över den kommunikativa aspekten. I nästa led kan det antas att den kommunikativa aspekten spelar roll. Ett exempel på detta kan vara en ekonom som bevakar räntor på lån för att minimera ett företags räntekostnader (Avdic 1995a:67ff). Anvecklaren/ekonomen bygger ett kalkylsystem för att bevaka ränteutvecklingen. Systemets utdatafunktioner ger information som kan föranleda åtgärder om ekonomen utifrån sin professionella bedömning anser så vara befogat. Den kommunikativa aspekten är i detta fall underordnad den informationsskapande. Ekonomens intentioner bör vara uppenbara för honom själv. Är de inte det kan det vara svårt att förmedla dem till någon annan. Skulle däremot utdata riktas till medarbetare eller ledning spelar ekonomens intentioner roll för tolkningen av de data som systemet genererar. Om t.ex. systemet varnar vid vissa gränsvärden, vika åtgärder skall då vidtagas och varför.

²⁰ "A communicative action model of business relations and business processes."

Kunskap



Figur 101 I anvecklarens praktik kan huvudpraktiken ge uppdrag till utvecklarpraktiken.

13.11 Alternativa ansatser

Problemet att närma verksamhet och IT är inte nytt. Det har sedan datorer började användas, funnits en önskan att utnyttja dem i verksamheter för att stödja dessa. Ett av hindren för att stödja verksamhet har varit svårigheter för systemutvecklingsspecialister att ta del av verksamhetsföreträdarens kunskap om verksamheter. Detta kan ses som uttryck för svårigheter med att artikulera tyst kunskap. Olika sätt att närma sig problemet har framställts under årens lopp. Användarvänliga systemutvecklingsmetoder har varit ett sätt. Utbildning av användare i systemutvecklingsmetoder har varit ett annat. Nämnda två exempel kan sägas vara initierade från systemutvecklingssidan snarare än från verksamhetssidan.²¹ I och med utvecklingen på programvaru- och maskinvarumrådet har möjligheter yppats att närma sig problemet mer från verksamhetssidan. Uttrycket och studieområdet *Computer Supported Work* kan ses som en ambition att sätta arbetsuppgifter i centrum med datorer som stöd. Ett annat verksamhetsorienterat synsätt presenteras genom Nurminens HIS-modell (Nurminen 1995:211) (se kap 9). Inom ramen för sin HIS²²-modell redovisar Nurminen (1988:112ff) en ansats som syftar till att integrera formaliserbar systeminformation²³ och icke-formaliserbar mänsklig information²⁴. HIS-modellen fokuserar på problemet med informationsöverföring via datorer. Enligt Nurminen kan t.ex. information inte överföras, utan att informationsavsändaren finns tillgänglig. Skälet är att information inte kan formaliseras fullt ut. Artificiell intelligens (AI) är ett annat försök att utnyttja datorstöd för att formalisera kunskap som är tyst (se vidare kap 12).

²¹ Förutsatt att man ser traditionell systemutveckling som en aktivitet med två huvudsakliga roller: verksamhetsföreträdare och systemutvecklingsföreträdare.

²² Human-Scale Information Systems.

²³ "Formal system- information".

²⁴ "Non-formal human information".

En delvis annan syn på formalisering av kunskap redovisar Göranson (1990), som diskuterar risken av att professionell kunskap försvinner då den automatiseras. Göranson menar att datorisering kan bryta ett viktigt samband mellan beräkning och bedömning, "det dubbla greppet". De system som gör att professionell kunskap riskerar att försvinna kan snarast jämföras med TSU-orienterade expertsystem. De anvecklade system som redovisas i *Del II*, tillkommer på utvecklarnas initiativ för att öka kvaliteten och ofta utveckla ny kunskap. Kravet på rimlighetsbedömningar och fördjupad kunskap understryks av samtliga respondenter. Kalkylsystemen i exemplen syftar till att befästa och fördjupa den professionella kunskapen, inte ersätta den (se även kap 14).

13.12 Sammanfattning - Kunskap och anveckling

En sätt att dela in kunskap är i typerna påståendekunskap, utförandekunskap och förståelsekunskap. *Påståendekunskap* kan betecknas som faktakunskap, utan att förden skull vara entydig eller lättformaliserad. Påståendekunskap är en form av kunskap, som traditionellt varit föremål för formalisering i informationssystem, speciellt i form av databaser. *Utförandekunskap* kan betecknas som kunskap om handlingar och hur dessa utförs. Även vissa former av utförandekunskap kan formaliseras i form av informationssystem. Beräkningsregler i program t.ex. kan representera vissa former av utförandekunskap. Andra former av utförandekunskap har varit föremål för formaliseringsambitioner inom området artificiell intelligens (AI), men är inte sällan av den karaktären att den är svår att formalisera. *Förståelsekunskap* är till sin karaktär sådan att den inte låter sig formaliseras på något entydigt sätt i ett informationssystem. *Personlig kunskap* tolkas här som påståendekunskap, utförandekunskap och förståelsekunskap. Den personliga kunskapen är till stora delar tyst.

För att formalisera påståendekunskap krävs *praktisk kunskap*. Systemutvecklare har utvecklat och utvecklar praktisk kunskap om systemutveckling. Delar av denna kunskap kan utvecklare behöva för att utveckla/anveckla. IT-specialisters praktiska kunskap kan överföras på olika sätt. Ett sätt är att utvecklarna lär sig tillvägagångssätt och principer om datastrukturering, programmering etc. Ett kompletterande sätt är att integrera IT-specialisternas kunskaper i utvecklingsprogramvara t.ex. kalkylprogram. Denna integration kan ske på så vis att färdigprogrammerade standardfunktioner helt eller delvis ersätter manuell programmering utförd av programmerare. Ett exempel på detta är pivottabeller i kalkylprogram²⁵, som ger möjlighet att skapa rapporter utan att programmera. Utvecklandet av motsvarande rapportrutiner skulle i slutet av 1980-talet krävt åtskilligt programmeringsarbete för programmerare och varit näst intill omöjligt att åstadkomma för en standardutvecklare.

I kapitel 1 uppmärksammades två typer av kunskap, (1) utvecklingsverktygskunskap och (2) verksamhetskunskap, där man kan säga att (1) kan ses som en delmängd av (2). För systemutvecklaren är utvecklingsverktyg en del av (kärn)verksamheten. Utifrån ovanstående kunskapsklassificering (påståendekunskap etc.) finns, när det gäller utvecklingsverktygskunskap påståendekunskap i form av kunskaper om t.ex. prestanda

²⁵ I MS Excel

Kunskap

hos olika datortyper, utförandekunskap i form av t.ex. programmeringskunnande samt förståelsekunskap i form av insikter och erfarenheter om t.ex. användarengagemang inom området systemutveckling. Verksamhetskunskaper varierar med verksamhetsområde.

Begreppet personlig kunskap framstår som relevant i relation till anveckling eftersom det rör sig om människor som arbetar i verksamheter, utför arbetsuppgifter och använder verktyg av olika slag för att lösa dessa uppgifter. Utifrån den pragmatiska kunskapssyn som beskrevs inledningsvis, är kunskap och handling integrerade hos anvecklare. ”Det betyder att man inte helt kan separera kunskap och handling.” (Goldkuhl 1997:6) Anveklares uppfattning om sina arbetsuppgifter och hur dessa skall lösas är personlig. Inom ramen för anveklarens arbetsuppgifter kan det förutsättas att det normalt finns behov av personlig kunskap. Denna kunskap är inte enkel att formalisera.

“When you want to turn knowledge to information and information into data you will often have to make explicit what has hitherto been tacit in the competence of those having knowledge. This can be a difficult task fraught with all kind of obstacles.” (Dahlbom & Mathiassen, 1991:32)

Anvecklare kan explicitgöra tyst kunskap genom att bygga kalkylsystem, vilket i sin tur gör den möjlig att värdera och kritisera för en vidare grupp intressenter.²⁶ I förlängningen kan detta ses som en del i ett kvalitetsutvecklingsarbete i en organisation.

Anveckling och kunskapsformer kan för det första relateras till varann genom att olika typer av utvecklingsverktygskunskap kan vara olika svåra/lätta att formalisera och därmed kommunicera till anveklaren. För det andra kan verksamhetskunskap vara olika svår/lätt att formalisera och därmed kommunicera till systemutvecklaren. Båda dessa faktorer påverkar både anveklarnas och systemutveklarnas möjligheter att utföra arbetsuppgifter och utveckla system (se även Figur 8). För systemutvecklaren är det nödvändigt att dels ha kunskap, som gör det möjligt att *verifiera* verksamhetskunskaperna och dels kunskap som gör det möjligt att *specificera* datasystemet. Anveklaren har normalt inte samma verifieringsbehov eftersom han/hon definitionsmässigt står för just verksamhetskunskapen.

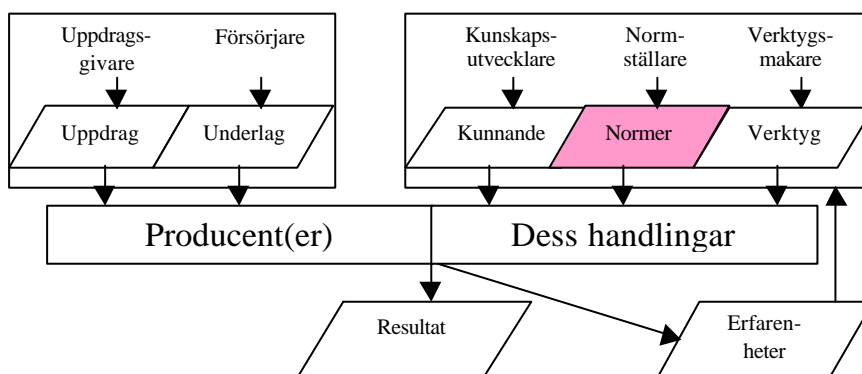
²⁶ Exempel på detta finns i Avdic (1995a:67ff) och i kapitel 3 och 4, t.ex. Bosses **täckningsbidrag per timme**, där uppfattningar om mätning av produktion konkretiserats i ett kalkylsystem.

Normer

"Den som inte älskar vin, kvinnor och sång
förblir en narr livet lång."
(Martin Luther)

- Syftet med detta kapitel är att relatera normbegreppet till anvecklingsbegreppet

14 Normer



Ett av delsyftena med denna avhandling är att med KPA "visa hur hänsyn kan tas till svårformaliserbar kunskap och svårformaliserbara mål." Med svårformaliserbara mål avses här yrkes- och verksamhetsetik (se kap 1.2.8), som tar sig uttryck i normer och regler (Collste 1996:76ff). Ett av flera exempel från D1 (Avdic 1995a:72) gäller en ekonomichef i ett allmännyttigt bostadsföretag, som utvecklade ett kalkylsystem för att simulera/beräkna hur hyran skulle kunna sänkas i olika trappuppgångar i utbyte mot att hyresgästerna utförde vissa sysslor som trappstädning, snöskottning och gräsklippning. Bakgrunden till dessa överväganden var företagets policy, som innebar en strävan efter lägsta möjliga hyreskostnader och strävan efter demokratiskt inflytande över boendekostnader. Med kalkylsystemet kunde policyn i viss utsträckning förverkligas. Exemplet belyser möjligheterna att med anveckling tillgodose svårformaliserbara mål i termer av verksamhetsetik (företagets policy).

Syftet med detta kapitel är att relatera begreppet *normer*¹ till anvecklarens praktik. I kapitel 7 introducerades begreppen *kvalitetsnormer* och *handlingsnormer*.

"Att utföra en praktik innebär en strävan efter att utföra denna praktik väl. ...För att göra detta behövs normer och regler för att bedöma handlingars och resultatets godhet." "Producenternas handlingsutrymme regleras av kvalitetsnormer och handlingsnormer. Kvalitetsnormer talar om vad som är bra och dåligt och handlingsnormerna talar om vad som får göras och vad som inte får göras." (Goldkuhl & Röstlinger 1997:8)

Kvalitetsnormer kan enligt detta sätt att se klassificeras som en typ av mer eller mindre uttalade rekommendationer medan handlingsnormer kan förknippas med konsekvenser i form av någon form av bestraffning för den som överträder dem. *Yrkesetik* och *verksamhetsetik*, som diskuterats i kapitel 1, kan normalt räknas som kvalitetsnormer, dvs rekommendationer som syftar till att uppnå någon form av kvalitet.² Begreppen yrkesetik och verksamhetsetik, som är centrala för anvecklaren introducerades som relaterade till verksamhetskunskap. Yrkesetik representerade normer relaterade till professionskunskap och verksamhetsetik representerade normer relaterade till verksamhetskunskap. Exempel på yrkesetik i anvecklingsverksamheten är Jan (se kap 4.4) som i sin ekonomichefsroll styrs av regler för kommunalekonomer (se Bilaga 6). I kapitel 4.4 redovisas hur reglerna påverkar Jans anvecklingsverksamhet. Exempel på verksamhetsetik i anvecklingsverksamheten är Olle på samma myndighet (se kap 4.3), vars verksamhets som exploateringsingenjör styrs av den kommunala verksamhetens mål (se 4.3.2).

Praktikförutsättningarna kunskap, normer och verktyg överlappar varandra (se kap 7, 13 och 15). En uppdelning i kapitel (som gjorts i denna avhandling) kan innebära att aspekter delvis behandlas dubbelt eller att relaterade aspekter behandlas åtskilt. Kunskap kan relateras såväl till normer som till verktyg liksom verktyg kan ses som formaliserad kunskap och formaliserade normer.

Normer och kunskap är alltså intimt förenade. Gränserna mellan yrkeskunskap och normer i ett yrke är flytande. Å ena sidan är kunskap inte sällan normativ. Å andra sidan är kännedom om normer ett slags kunskap. En kocks kunskaper om matlagning innehåller med automatik en slags normer. "Timjan passar bra till stekta kycklingrätter" är en norm såtillvida att den uttrycker en värdering. Å andra sidan kan den också ses som en form av kunskap. Omständigheten att normer och kunskap är intimt förknippade innebär inte att de är likvärdiga. Påståendekunskap kan vara sådan att normer kan vara svåra att upptäcka. Att känna till att ett kalkylblad har ett visst antal rader är kunskap men knappast normer. Att se till att ett kalkylsystems funktion inte begränsas av ett kalkylblads storlek kan däremot ses som en kvalitetsnorm, som bygger på en

¹ "se *rättesnöre*, lag, ka'non, grundregel, princip, ledande synpunkter; prototyp; (ibl.) mått, måttstock, likare, standard, schablon, modell, kliché, formel." (*Ord för Ord*, 1964:441) Under *regel* står i samma bok: "rättesnöre, föreskrift, stadga(nde), (teol.) dogm, ka'non, ;formel, *förordning*, norm, system, plan, schema, metod;..." Begreppen norm och regel används här synonymt.

² Undantag skulle kunna vara professionssammanslutningars etiska regler, som om de bryts mot, kan medföra uteslutning ur samfundet, t.ex. advokatsamfundets etiska regler.

form av påståendekunskap. Om systemets databas riskerar att överträda maximala antalet rader kan kalkylbladets storlek även ses som en handlingsnorm.³

14.1 Normer och mål

Normer och mål är två relaterade begrepp.

"Mål kan ses som någon typ av värdeangivelse som anger vad man vill uppnå med verksamheten. Målet anger ett resultat eller ett önskat läge/tillstånd." (Goldkuhl & Röstlinger 1988:25)

Definitionen av norm enligt Nationalencyklopedin anger en besläktad betydelse

"...handlingsregel, påbud om hur något bör vara beskaffat eller organiserat. Partikulära normer gäller enskilda situationer; generella normer kan tillämpas i en mängd olika fall. Inom moralfilosofi och politisk filosofi studeras huvudsakligen generella normer för moraliskt riktigt handlande eller för en riktig organisation av samhällen, t.ex. utilitarismen, Kants kategoriska imperativ och John Rawls principer om rättvisa. Enligt en vanlig uppfattning kan normer inte logiskt härledas från beskrivningar av fakta. Många anser dessutom att normer varken är sanna eller falska." (Nationalencyklopedin)

Mål i en verksamhet kan vara av olika karaktär. Goldkuhl & Röstlinger (1988:86) gör följande indelning av mål (Tabell 19 nedan):

Tabell 19 Målkarakterisering (Goldkuhl & Röstlinger 1988:86).

Målets karaktär
Huvudmål – delmål
Giltighetsområde
Officiellt
Dokumenterat
Känt
Accepterat
Verksam
Uppfyllt
Verklighetsanpassat
Bidragande – motverkande
Mätbart
Tidsberoende – tidsberoende

Karakteriseringen ovan skulle även kunna gälla för normer. Normerna i *Bilaga 5* (ACM:s Code of professional conduct) är ett exempel på sådana normer. Därmed sägs inte att alla mål är normer. Om t.ex. en höjdhoppare har som mål att klara 2 m och 30 cm ett visst år, är inte det att betrakta som en norm. Likaså om ett företag har som mål att uppnå en specifik avkastning på eget kapital. I de två exemplen kan målen nås på olika sätt och målet i sig säger inget om hur det skall nås. Målen innehåller inga

³ I MS Excel är maximala antalet rader på ett kalkylblad 65536.

rättesnören, inga normer. Vissa mål kan sägas gälla för verksamheter medan etiska regler gäller professioner i verksamheter. Normer kan vara förknippade med mål i form av restriktioner för hur målet skall nås, dvs som ett slags regler. Mål i verksamheter kan vara mer eller mindre (uttalat) relaterade till normer. Att uppfylla normer kan sägas vara ett mål i en profession, se t.ex. *Bilaga 5* (ACM:s Code of professional conduct). Etiska regler kan förekomma i en profession utan att vara relaterade till något verksamhetsmål.

"Strävan efter det bästa är inte en strävan efter ett i förväg givet mål, det är en inre riktningssvisare i kunskap i handling." (Molander 1996:80)

Det är inte uteslutet att normer i form av yrkesetik och verksamhetsmål kan vara motsatta. Ett exempel på det kan vara en läkare i ett idrottslag som ger en nyckelspelare spelförbud inför en viktig match. Etiken i läkarprofessionen kan stå i motsättning till målen med idrottsverksamheten. Normer för utvecklare finns i kapitel 17.2 *Det goda kalkylsystemet*.

14.2 Normer i relation till skill, know-how och kompetens

Rolf (1991:111) menar att praktisk kunskap i form av handlingar kan utföras på olika sätt och att normer används för att värdera om handlingen är bra eller dålig. I kapitel 13.8 diskuterades tre former av praktisk kunskap, skill, know-how och kompetens. Det som skiljer typerna är hur det kan värderas om handlingar är bra eller dåliga. För skill-kunskap kan aktören själv värdera handlingar (om man t.ex. inte kan simma). För *know-how* gäller att "det sociala kollektivet" bidrar med normer. Normerna kan vara internaliserade så att handlingar utan reflexion anpassas till dem eller så kan reglerna vara externaliserade i form av t.ex. etiska regler för en profession. Om know-how vilar på situationsberoende måste ett kontinuerligt lärande äga rum. Om det i en organisation sker ett omprövande av verksamhetens kvalitetsmål och inriktning, måste aktörer i organisationen kontinuerligt lära sig dessa regler för att kunna anpassa sig till dessa.

Rolf (1991:121) definierar en aktör med *kompetens* på följande sätt:

"En aktör A har kompetens inom området O =

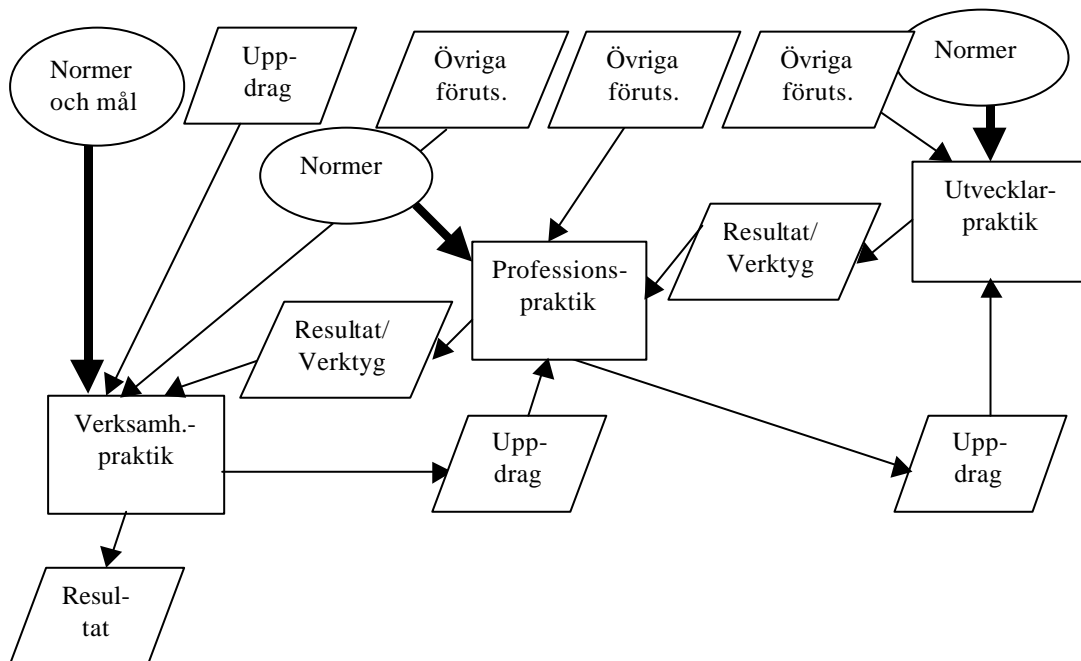
- a) Det finns på området vissa regler eller kriterier för kvalitet som skiljer en väl utförd handling från en sämre eller dåligt utförd handling.
- b) Dessa regler eller kriterier är förankrade bland aktörer på området. Deras bedömningar avgör vad som utgör bättre eller sämre prestationer.
- c) Aktören har förmåga att handla i enlighet med dessa regler genom sin tysta kunskap om dessa kriterier eller regler.
- d) Aktörens förmåga innehåller hantering av symboliserande, teoretiska, reflekterande eller kritiska moment.
- e) De reflekterande momenten kan aktören använda för att på ett förutsebart sätt påverka kvalitetskriterierna inom området.
- f) Denna påverkan gör reglerna bättre - enligt vissa latent regler - än vad de annars skulle ha varit."

Enligt definitionen på kompetens skall en aktör kunna påverka norm/regelsystemet för att kunna sägas ha kompetens. Att enbart följa det socialt relaterade befintliga norm/regelsystemet klassificeras enligt Rolfs indelning som know-how. En person kan

Normer

ha relation till flera regelsystem och "...ha kompetens förhållande till ett regelsystem men sakna kompetens i förhållande till ett annat." (Rolf 1991:122) När Jan tillverkar ett budgetsystem med kalkylprogram (se kap 4.4.4), har han relationer till minst tre normsystem (se Figur 102 nedan). Normsystemet i huvudpraktiken, den kommunala förvaltningen, är detsamma som verksamhetsetik (se kap 1.2.7) och kan vara formulerat i t.ex. verksamhetens policy. Normsystemet i den ekonomiska professionen är det samma som yrkesetik (se kap 1.2.6), och kan vara formulerat i någon form av regelverk för ekonomer (se exempel i *Bilaga 6*). Normsystemet i utvecklarpraktiken kan vara mer eller mindre obekant för ekonomen. Observera att praktikerna kan ses som tre aspekter av anvecklarens arbete (se sid 148).

Enligt definitionen på standardanvecklare har denne viss kunskap om utvecklingsverktyg, vilket medför att det inte är fråga om kompetens i relation till normsystemet i anvecklarpraktiken. Anvecklaren kan däremot ha kunskap och kompetens i sin verksamhetspraktik eller i sin professionspraktik (de kan också vara överlappande, se kap 3.3). Dessa kunskaper och normer kan vara mer eller mindre formaliserbara eller artikulerbara (Rolf 1991:100). Enligt syftet med denna avhandling är syftet att visa hur hänsyn kan tas till svårformaliserbar kunskap och svårformaliserbara mål, dit verksamhetsetik och yrkesetik kan räknas.



Figur 102 Olika normer gäller i anvecklarpraktikens olika (del)praktiker.

I anvecklings-sammanhang förekommer följaktligen normer i minst två praktiker, i yrkespraktiken och i anvecklarens utvecklarpraktik. Normer i yrkespraktiken är knutna till professionen, medan normer i utvecklarpraktiken är knutna till systemutvecklingsaktiviteterna i utvecklarpraktiken. Flera av de problem som uppmärksammas i relation till anveckling kan i själva verket förklaras med bristande kunskap om systemutveck-

lingsnormer (se t.ex. kap 10). Jan uttrycker även en stark önskan att tillägna sig mer kunskaper för att kunna bygga system (se kap 4.4.5).

14.3 Mål- och värdebaserad praktik

En aspekt av normer som behandlas i kapitel 7 är skillnader mellan *procedurbaserad* och *målbaserad praktik*. När det gäller målbaserade praktiker skiljs på *operativa-målbaserad praktik* och *strategiska-mål-baserad praktik*. Praktiktyperna kan relateras till olika handlingstyper, vilka i sin tur resulterar i olika typer av normer. Habermas (1995) beskriver⁴ fyra typer av handlingar: målrationala, värderationella, affektiva och traditionella. Handlingstyperna beskrivs kortfattat i Tabell 20 nedan.

Tabell 20 Aspekter av handlingar och deras innebörd (Habermas 1996:98).

Handlingstyp	Beskrivning
Målrational	Bestäms av att aktören uteslutande strävar efter att uppnå ett preciserat mål.
Värderationell	Bestäms av en tro på t.ex. det etiska egenvärdet hos ett beteende som sådant och oberoende av dess resultat.
Affektuell	Bestäms av aktörens känslotillstånd.
Traditionell	Bestäms av vanor.

Indelningen i Tabell 20 ovan ser endast till vissa karaktäristiska drag. Detta innebär att en renodlat målrational aktör endast ser till mål och inte bryr sig om normer. En renodlat värderationell aktör ser idealtypiskt endast till normer och inte till mål. En idealtypisk utvecklare kan följaktligen framstå som orealistisk.

Målbaserade praktiker (operativa-mål-baserade) ger större handlingsutrymme för utvecklaren vad gäller tolkning av normer, uppdrag, behov av kunskap och verktyg etc. Med målbaserad praktik följer *målbaserade normer*. Att normer är målbaserade innebär att handlingars kvalitet kan avgöras i och med att man uppskattar i vilken grad de bidrar till måluppfyllelse. Målbaserade normer kan ha sin grund i rationella nyttoargument. Exempel på målbaserade normer är t.ex. att en verksamhet skall ge ett visst täckningsbidrag för att en viss lönsamhet (målet) skall uppnås. Exempel på en målbaserad norm från IT-området är att databaser skall vara normaliserade för att inte bli inkonsistenta (målet).

Värderationella handlingar kan relateras till strategiska-mål-baserad praktik. Värderationella handlingar styrs av normer. Ett exempel på det är läkarprofessionen, vars verksamhet till stor del har byggts på normer om människans rätt till liv ända sedan Hippokrates tid. Strategiska mål kan också ha karaktären av normer. Om ett företag har en policy av typen "hög produktkvalitet och hög kundservice" så har dessa normer en värderationell karaktär. En skillnad kan vara att läkarprofessionens normer normalt inte underordnas målbaserade normer utan vidare. Strategiska-mål-baserade normer kan dock underordnas mål i en verksamhet eller åtminstone förändras om t.ex. löns-

⁴ Med utgångspunkt från Weber.

samhetsmål står i motsatsställning. Strategiska mål-normer är en slags kvalitetsnormer. Kvalitetsnormer är i sig ett slags mål. Om t.ex. ett allmännyttigt bostadsföretag strävar efter att öka hyresgästinflytandet (Avdic 1995a:67ff), så är det ett mål och dessutom en norm som gör det möjligt att värdera handlingars kvalitet utifrån.

Procedurbaserade normer innebär att proceduren framstår som ett mål i sig. Kultur- eller traditionsbaserade normer kan vara procedurbaserade. Exempel på det är regler som t.ex. omgärdar servering av måltider. Skälet till varför man skall lägga bestick på ett visst sätt eller äta olika rätter i en viss ordning kan ibland uppfattas som relativt ogrundat. Det ursprungliga skälet kan måhända ha varit rationellt men har med tiden försvunnit i glömska och endast efterlämnat en kulturellt betingad regel.⁵ Detta innebär att normer som är nyttorelaterade (målbaserade) kan förlora sin nyttoas och förvandlas till kulturellrelaterade normer. I avsaknad av mål att värdera handlingar mot kräver procedurbaserad praktik väldefinierade procedurer. Om målen inte finns artikulerade eller har fallit i glömska kan argumenten ibland formuleras: "Så har vi alltid gjort!" Denna inställning kan vara en grund för anvecklarens analys av sin verksamhet, vilket framförs av Ismo i kapitel 3.3.

I kapitel 1 introducerades betydelsen av mål i anvecklarens verksamhet i den bemärkelsen att målen kunde vara verksamma utan att behöva formaliseras. Den styrning som *hög-kontroll-synsättet* förordar (se kap 10.4.7) är närmast relaterad till procedurbaserad praktik. Normerna behöver inte vara nyttorelaterade, man styr ändå utifrån definierade procedurer. *Låg-kontroll-synsättet* är närmast relaterat till målbaserad praktik. Normerna bör vara nyttorelaterade om man vill att verksamheten skall styras mot mål.

Göranzon (1990) beskriver erfarenheter av datorisering av skogsmästares arbete med att värdera skogsfastigheter. Vid en datorisering av beräkningsarbetet i värderingsarbetet uttrycks farhågor för att inte bara beräkningskunskapen skall gå förlorad utan även förmågan att värdera kvaliteten på värderingen. Göranzon menar att förmågan att bedöma ett resultat hänger samman med skogsmästarnas arbete med att beräkna och bearbeta ett material. Göranzon menar att "...beräkningsförmågan och bedömningsförmågan är två sidor av samma sak. - Det dubbla greppet." (Göranzon 1990:67) Skogsmästarnas handräkning var viktig för yrkesutövandet.

"För det första rörde det sig inte om ett mekaniskt räknande. Räknandet var ständigt interfolierat av rimlighetsbedömningar. Det var en helhet av beräkning och bedömning. Det går inte att dra någon skarp gräns mellan rutinmässiga och kvalificerade moment."

"En andra punkt är räknandets karaktär av *fördjupning i materialet*, om man vill en *reflektion* över det man fått fram vid inventeringen av fastigheten. Det är genom att skogsmästaren räknar på materialet - och först då som han får *överblick*." (Göranzon 1990:68)

Studien visar att normer upprätthålls genom bearbetning av (siffer)material. En alltför långtgående automatisering av utförandekunskap kan innebära mindre reflektion och överblick och därmed mindre möjlighet till rimlighetsbedömningar.

⁵ På idrottsområdet finns gott om kulturellrelaterade regler.

Molander (1996) reflekterar i termer av målrationalitet över Göranzons studier av datorisering av arbetsliv.

”Det dubbla greppet ger också insikt om faror med en kunskapssyn som domineras av ett målrationalt perspektiv, dvs att *givna* mål skall uppnås med effektiva medel, instrumentell kunskap.” (Molander 1996:224f)

Exemplet med skogsmästarna visar att inte bara kunskap är nära relaterat till handling utan också normer.

I de exempel som redovisas i kapitel 3 och 4 innebär anvecklares användning av kalkylprogram snarast att system utvecklas för att upprätthålla och fördjupa det professionella kunnandet och verksamhetskunskapen. För anvecklarna sker anveckling för att fördjupa och verifiera kunskaper om verksamheten i enlighet med uppfattade mål. Vid intervjuerna var just dessa aspekter starkt framträdande. Exempel på kalkylsystem där detta visas är t.ex. **kostnadsjämförelse för arkning** (se sid 69), **uppföljning av arkmaskiner** (se sid 83), **tomtprisberäkningssystem** (se sid 94) och **markvärdering** (se sid 98). En gemensam egenskap hos de studerade anvecklarna är deras ambition att söka korrekt och noggrann kunskap. De beskrivna systemen är ett uttryck för detta.

14.4 Normer, kunskap och yrkesetik i professioner

Att kalkylprogram kan bidra till att anvecklare uppfyller svårformaliserbara mål i form av t.ex. yrkesetik är en bärande idé i denna avhandling. Av detta skäl behandlas nedan begreppet yrkesetik och dess relation till kunskap (och verktyg). Ett exempel på detta är hyresberäkningssystemet (Avdic 1995a:72) som nämnts ovan. Ett annat exempel från samma företag gäller en finansieringsekonom, som ansvarade för att den omfattande låneverksamhet, som finansierade företagets byggprojekt skulle ge så lite onödiga förluster som möjligt. Yrkesetiska hänsyn genomsyrade varje penningplacering på detta område, där lagar inte alltid är entydiga och lättolkade. I *Del II* finns fler exempel på detta, t.ex. i kapitel 4.4.

Normer, kunskap och yrkesetik är besläktade begrepp. Yrkesetik kan betraktas som normer i en profession. Enligt den praktikgeneriska modellen finns en överlappning mellan normer och kunskap. Såväl normer som kunskap kan även ses som inbyggda i verktyg.

”Verktyg som fungerar och är effektiva alstrar arbetsglädje och verksamhetslust. Men de rymmer också insikter.” (Molander 1996:14)⁶

Yrkesetik kan förekomma i olika typer av yrken. De professioner som kan tänkas utnyttja kalkylprogram för att utföra arbetsuppgifter i enlighet med sin yrkesetik kan vara av skiftande slag. Administratörer, ekonomer, tekniker är några övergripande kategoriseringar där kalkylprogram (eller andra anvecklarverktyg) kan förekomma som verktyg. De yrkesetiska normerna styr i dessa professionstyper anvecklarens ställningstaganden i hans arbete.

⁶ Med referens till Tempte.

Normer

"Etiken finns i handlingen, i varje ställningstagande – den styr uppmärksamheten." (Molander 1996:18)

För t.ex. den finansieringsekonom som nämndes ovan kan valet av lånetyper och finansieringsalternativ innebära en balansgång mellan risktagande och avkastning. Ethiska regler kan styra beslutsfattandet (Avdic 1995a:71f).

Yrkesetiken i en profession är en del av den professionelles yrkesroll. Det är medlemmarna i professionen som värnar om yrkesetiken. "Centralt i överföring av yrkestradition är yrkesetik." (Molander 1996:14). En skillnad mellan olika professioner är att yrkesetik kan vara olika formaliserad. Vissa professioner kan ha en dokumenterad normsamling, t.ex. FAR (för revisorer) och ACM Code of professional conduct (för IT-professionella). I andra professioner kan yrkesetiken vara underförstådd och finnas i form av tradition och yrkesstolthet. Exempel på sådana professioner skulle kunna vara produktionsplanerare och lärare. Yrkesetik kan vara svår att generalisera mellan professioner.

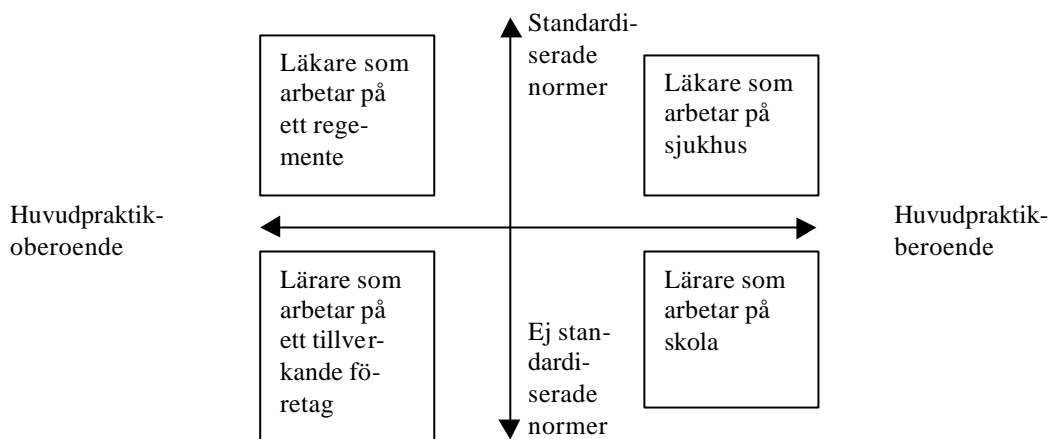
"En hantverksetik i överförd bemärkelse handlar för mig mer om villkor för arbetets ära och utövning." (Tempte 1997:82)

Yrkesetik handlar om att handla och sträva efter det bästa resultatet i relation till de yrkesetiska normerna.

"Etiken finns i arbetet, i handlandet, i uppmärksamheten, i ställningstagandena. Yrkeshedern finns i kunnandet såsom det utövas. Yrkeskunnandet är inte ett »instrument» för ett mål som kan definieras oberoende av verksamheten. Kunskap och etik – alltså »strävan efter det bästa» – står i ett inre samband med varandra." (Molander 1996:80)

Enligt den vida definition av professionsbegreppet, som ligger till grund för denna avhandling innefattas även yrken, som inte är starkt standardiserade, i professionsbegreppet. Det faktum att olika professioner är olika standardiserade påverkar dock hur den professionelle uppfattar normer i sin verksamhet. För advokater och läkare kan yrkesetiken finnas väldokumenterad och vara allmänt accepterad och institutionaliserad. För professioner inom t.ex. IT-området kan normsystemen vara olika dokumenterade. Inom andra områden, t.ex. produktionsplanerarens kan yrkesnormer vara beroende av verksamheten. Standardiserade normer kan sammanfalla med verksamhetsmål, exempel på det är yrkesetiken för läkare som arbetar på sjukhus. Yrkesetiken för t.ex. regementsläkare kan däremot upplevas som delvis motsatt verksamhetsmålen, åtminstone av värnpliktiga. Det beskrivna förhållandet illustreras i Figur 103 nedan. Normer kan vara verksamma i en profession även om de inte är nationellt eller internationellt standardiserade. Den geografiska utsträckningen på normerna kan variera, men de kan fördensskull vara verksamma för den professionelle i den verksamhet, som han verkar i. Exempel på detta kan vara lärare, lantbrukare eller produktionsplanerare. Även mindre standardiserade normer kan vara mer eller mindre relaterade till den professionelles huvudpraktik, vilket illustreras i Figur 103 nedan, där lärare får tjänstgöra som exempel på en profession med mindre standardiserade normer. Då lärarens profession utövas i en skola förutsätts skolans mål och lärarprofessionens normer harmoniera, me-

dan detta inte behöver vara fallet om läraren arbetar i en verksamhet som sysslar med annan verksamhet än utbildning.



Figur 103 Exempel på professionsnormer i relation till huvudpraktik och standardisering.

Att verksamhetsmål och professionsnormer inte samvarierar behöver inte betyda att de står i motsats till varandra, även om det är möjligt. Anvecklare skulle t.ex. kunna bättre uppfylla normerna både i sin professionspraktik och i sin huvudpraktik om de utövade normer från IT-specialistpraktik.

Viss skillnad vad gäller yrkesetik finns mellan teoretiska och praktiska typer av yrken. "En skillnad mellan teoretisk och praktisk kunskap är mycket tydlig: Den teoretiska är hela tiden förändringsbar och osäker, den är inte definitiv och kan aldrig vara slutgiltigt rätt eller fel. Praktikens kunskap är definitiv och påtaglig." (Tempte 1997:82)

När Tempte talar om praktisk kunskap menar han här hantverk, t.ex. möbelsnickare och det definitiva ligger i det faktum att när ingreppet i träet är gjort kan det inte återskapas. Temptes uppdelning är dock inte kategorisk. Det finns teoretiska professioner, vars kunskap kan ha en definitiv karaktär. En finansieringsekonom kan placera ett lån och binda det. En produktionsplanerare kan planera produktion, som inte kan göras ogjord. Det är i dessa fall yrkesetiken kommer in. Det är de yrkesetiska reglerna som vägleder ekonomen och produktionsplaneraren i sina handlingar för göra det bästa valet. Användning av kalkylprogram kan hjälpa anvecklaren att leva upp till sin yrkesetik. För Ismo i kapitel 3.3 kan kalkylsystemet **kostnadsjämförelse för arkning** fungera på detta sätt. Beräkningarna kan leda till omprioriteringar i syfte att anpassa verksamheten till befintliga mål.

Till praktisk kunskap i en profession är begreppet ansvar väl relaterat. Yrkesetik innebär i sig ett ansvar att leva upp till normer. Den professionelle har ett val och han kan välja att inte leva upp till normerna. I begreppet professionalism ligger att konsekvenserna blir olika beroende på hur yrkesetiken följs.

"Till levande kunskap hör också ett ansvar gentemot andra, i första hand detta: om jag gör anspråk på att ha kunskap så har andra rätt att lita på mig, att lita på att jag kan göra det jag säger eller på andra sätt låter förstå att jag kan. De har rätt

att lita på produkter, omdömen, råd och så vidare – kunskap i handling hör hemma i en *helhet*." (Molander 1996:250)

Ansvarsbegreppets har en central betydelse för anveckling. Det spelar roll hur anvecklaren handlar i sin praktik. I definitionen av anveckling ligger att anvecklaren har ansvar för hela eller delar av verksamheten. Ett ansvar för en anvecklare är knutet till mål och normer. Anvecklarens ansvar för sin verksamhet, upplevdes som en stark drivkraft bakom anveckling i de studerade fallen.

14.5 Förändring av normer

Argyris & Schön (1996:21) kallar den process där lärande påverkar den använda kunskapen (theory-in-use) som implicit finns i organisationen för *double-loop learning*.⁷ Med double-loop learning menas inte enbart förändring i handlingsstrategier eller grundläggande antaganden bakom strategier (single-loop learning). Med double-loop learning avses förutom dessa förändringar även en förändring i normer. Förändringen kan enligt Argyris & Schön utföras av individer eller av organisationer som en följd av *undersökningar*.⁸ Med undersökningar avses upptäckt och korrigerande av felaktigheter, som uppträder i organisationer engagerade i transaktioner med sin omgivning. Då organisationens normer förblir oförändrade kan single-loop learning ses som en strävan efter effektivitet där verksamheten strävar efter uppfyllande av befintliga mål och normer. I de fall som korrigerande av felaktigheter leder till förändring av normer kan det benämnas double-loop learning.

Enligt definitionen av kompetens ovan kan endast personer med kompetens ha möjlighet att åstadkomma en förändring av normer i en organisation. De undersökningar som leder fram till korrigerande av felaktigheter måste ha en konkret form. I ett KPA-sammanhang kan anvecklares användning av kalkylprogram ses som en sådan typ av undersökning. Det analysarbetet som utförs av ekonomer, ingenjörer och andra kan leda till såväl single-loop som double-loop learning. Ett exempel på hur normförändring kan ske är Bosses **täckningsbidrag per timme** (se sid 77) där ett nytt sätt att mäta produktion påverkar företagens normsystem. I det nämnda fallet ledde KPA-analyserna till att det befintliga sättet att mäta produktion kom att ifrågasättas till förmån för en mätmetod som framstod som mer ändamålsenlig. Förändringen i mätmetod medförde förändringar i synen på verksamheten. Verksamheten kom att framstå i en ny dager i och med att andra verksamhetsmått kunde styra verksamheten. Även Thomas analysarbete med **uppföljning av arkmaskiner** syftade till att skapa en ny syn på bedrivande av verksamheten, vilken skulle ha kunnat leda till omvärdering av hur verksamheten skulle bedrivas.

14.6 Sammanfattning - Normer

Normer kan relateras till svårformaliserbara mål i termer av yrkes- och verksamhetsetik. Två typer av normer är kvalitets- och handlingsnormer. Normer och kunskap är i

⁷ Översätts av Rolf med om "en andra ordningens inläring".

⁸ "Inquiry".

anvecklingsammanhang två överlappande begrepp. Normer och mål är relaterade begrepp. Ett mål är en lägesangivelse som inte alltid ger vägledning om hur målet skall nås medan normer kan sägas ge viss sådan vägledning. Ett mål kan vara att uppfylla vissa normer.

Rolfs (1991) definition av kompetens innefattar den professionelles möjlighet att påverka professionens normer, vilket kan sägas vara relevant i anvecklingsammanhang. Olika normer gäller för anvecklarens olika praktiker. I anvecklarens utvecklingspraktik gäller IT-professionens normer. I mål- och värdebaserade praktiker spelar normer en central roll och är förknippat med styrning enligt *låg-kontroll-synsättet*.

Yrkesetik kan vara mer eller mindre relaterat till en anvecklares huvudpraktik. Huvudpraktikens karaktär kan påverka normer i termer av yrkesetik. I teoretiska yrken kan vad som är rätt och fel vara mindre påtagligt än i praktiska där konkreta handlingar omedelbart visar konsekvenser av handlande.

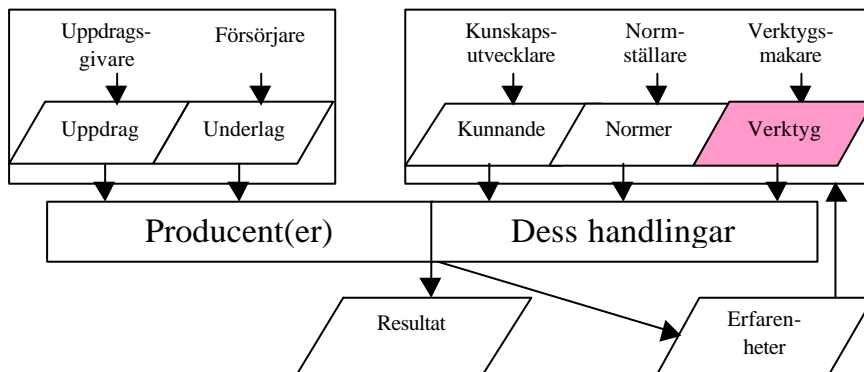
Det som Argyris & Schön kallar för *double-loop learning* innebär förändring av en organisations normer till följd av *undersökningar*. Detta kan relateras till anvecklingsverksamhet, när anvecklares analys av verksamheter innebär förändrade normuppfattningar i organisationen.

Utvecklingsverktyg

"**verktyg**, hjälpmedel för arbete av något slag. Många praktiska yrken har sina speciella verktyg. *Verktymaskiner* bearbetar arbetsstycken med verktyg t.ex. vid borrar, slipning, svarvning och sågning."
(Bra Böckers lexikon)

- Syftet med detta kapitel är att relatera verktygsbegreppet till anvecklingsbegreppet

15 Utvecklingsverktyg



Enligt den praktikgeneriska modellen är normer, kunskaper och verktyg relaterade (se kap 7). Ett verktyg kan innehålla formaliserade normer och formaliserad kunskap. Normer och kunskap kan också gälla verktyg. Normer kan gälla vilka verktyg som skall användas, hur dessa verktyg skall användas och av vem. Kunskap om verktyg avgör också val av verktyg och hur verktyget kan användas. Denna avhandling identifierar anvecklaren som en person som besitter goda kunskaper vad gäller verksamhet och viss kunskap vad gäller utvecklingsverktyg. Ju större kunskaper om utvecklingsverktyg anvecklaren har desto större kan möjligheterna vara att tillgodose behov och förverkliga möjligheter i verksamheten. Denna utgångspunkt placerar verktygsbegreppet i fokus för analys. Begreppet *verktyg* täcker in en mängd fenomen och kan innefatta företeelser från snörstumpar och grävmaskiner till metoder och abstrakta modeller. Kapitlet syftar till att analysera verktygsbegreppet utifrån ett anvecklingsperspektiv. Utvecklingsverktyg och deras potential är särskilt intressanta eftersom syftet med avhandlingen är att identifiera olika former av verktygskunskap och se hur denna kan bidra till att uppfylla anvecklares och verksamheters mål.

Abstrakta verktyg som metoder eller modeller är inte lika uppenbart något annat än enbart kunskap och normer. Då dessa metoder och modeller implementeras i programvaror (t.ex. kalkylprogram) blir de, om inte konkreta i bemärkelsen materiella, så i alla fall formaliserade. Gränsen mellan kunskap, normer och metoder blir uppenbart flytande då det gäller att identifiera metodverktyg för anveckling. Vad är en metod? Enligt Ord för ord (1964) betyder metod planmässighet, uppläggning, plan, regler, princip, skola. En tolkning av innebörden i dessa synonymer kan leda till en ganska vid betydelse av begreppet metod med innebörden *strukturerat tillvägagångssätt*. Metoder, som i denna bemärkelse, kan bidra till att en utvecklare kan utföra sina arbetsuppgifter kan vara implementerade i programvaror, t.ex. i kalkylprogram i form av guider eller andra styrda funktioner. Panko (1988:170) identifierar en form av metodkunskap som kan vara aktuell för utvecklare (se kap 13.3). Under rubriken *Data management* räknar Panko in följande regelområden:

- Fil-, katalog- och enhetsorganisation
- Förvaltningshantering, t.ex. backup och recovery

Under rubriken *Application development* räknar Panko in områdena

- Design
- Utveckling
- Dokumentation

Inom dessa områden skulle sålunda metodkunskap för utvecklare kunna finnas. Då denna typ av metoder, som ej är implementerade i verktyg ej identifierats i de empiriska studierna behandlas de ej utförligare i denna avhandling.

15.1 Verktygsbegreppet

Bra Böckers lexikon definierar *verktyg* på följande sätt:

"Hjälpmiddel för arbete av något slag. Många praktiska yrken har sina speciella verktyg. *Verktymmaskiner* bearbetar arbetsstycken med verktyg t.ex. vid borrar, slipning, svarvning och sågning." (Bra Böckers lexikon)

Uttrycket verktyg används i många sammanhang. Då verktyg definieras som *hjälpmedel* kan praktiskt taget allt vara ett verktyg. Inom industrin används t.ex. ordet verktyg som benämning på alla föremål, som används vid utförande av en viss arbetsuppgift. Det kan vara allt från en snörstump till ett arborrverk. Ett verktyg behöver inte vara konkret. Rosing (1978:38) menar att språket är ett verktyg för forskaren. Inom IT-området talas det ofta om verktyg. Verktyg inom IT-området kan vara programvaror av något slag. En kompilator eller en interpretator är ett verktyg, en databashanterare likaså. Som Nurminen (1995) uttrycker det:

"Information Technology, like any other technology, is a tool, a means, an instrument or a medium for accomplishing some task(s) of the primary work." (Nurminen 1995:215)

Pelle Ehn diskuterar i sin bok "Work-Oriented Design of Computer Artifacts" (Ehn 1988) huruvida en dator är ett verktyg. I sin definition av begreppet verktyg konstaterar Ehn (1988:393f):

- Verktyg är skapade av människor.

- Verktögsideal är "transparens"¹ och "konvivialitet"².

Ehn (1988:396) skiljer på verktyg och maskiner. Skillnaden ligger i oberoendet mellan operationen som utförs och yrkesskicklighet och motivation hos operatören. Verktyg låter sig manipuleras medan maskiner utför automatiserade handlingar. Enligt Ehns definition är ett kalkylprogram ett verktyg. Det är också ett verktyg med vars hjälp andra verktyg kan skapas, kalkylsystem. Ett kalkylsystem blir tillsammans med ett kalkylprogram ett verktyg för att skapa produkter i form av rapporter, diagram eller andra utdata. Att man med ett verktyg kan skapa ett annat verktyg är inget speciellt för dataområdet. För att göra en svarv behövs en svarv och en fräs kan inte byggas utan att man använder en fräs. Det som skiljer ett kalkylsystem från t.ex. en hammare är att de verktyg och maskiner som behövs för att skapa hammaren inte behövs när hammaren används, medan kalkylprogrammet (med vars hjälp kalkylsystemet skapas) ingår som en del av kalkylsystemet. (Delar av) verktyget i led ett är en delmängd av verktyget i led två.

Det som enligt ovanstående synsätt är speciellt med kalkylprogram-kalkylsystem i motsats till exempelvis hammaren, är att vissa kalkylsystem kan tendera till att bli maskiner medan ett kalkylprogram i sig alltid är ett verktyg. Ett kalkylsystem kan i likhet med ett informationssystem automatisera en del av en verksamhet så att det varken behövs motivation eller yrkesskicklighet för att använda det. Det kan till och med exekveras vid ett givet förutbestämt klockslag. Icke-styrda kalkylsystem däremot ger större möjlighet för användaren att använda kombinationen kalkylprogram-kalkylsystem som ett verktyg. Exempel på detta ges i kapitel 3 och 4.

Sett ur ett anvecklingsperspektiv finns flera programmerbara och standardiserbara programvaror som kan betraktas som möjliga för en anvecklare att använda. Enligt Holmberg (1995:52f) har databashanteringsprogram och kalkylprogram en liknande funktionalitet. Beekman (1997:69;73) skiljer på program för systemutveckling och andra applikationer. Till de program, som inte är till för (traditionell) systemutveckling, räknar Beekman ordbehandlingsprogram, desktopprogram, kalkylprogram, statistikprogram, bildhanteringsprogram, multimedieprogram, databashanteringsprogram samt kommunikationsapplikationer (Internetrelaterade program som e-post och webb-läsare). Det som gör att en typ av program kan vara lämpligt för anveckling kan bero på hur anvecklaren uppfattar verktygets egenskaper (Rivard & Huff 1988:557). Carlsson (1993:227) menar att verktygets inbyggda metafor skall passa anvecklaren. Andersen (1994:351) påpekar att användarledd systemutveckling bäst kan utföras med tillgång till "standardsystem eller fjärde generationens utvecklingsverktyg."³ Ett fjärde generationens verktyg är enligt Andersen ett verktyg som innehåller:

- Funktioner för skärmbilds- och dialogutformning
- Funktioner för databashantering
- Behandlingsspråk
- Rapportgenerator, frågespråk och grafiska funktioner

¹ Verktöget skall vara "...an extension of the body" (Ehn 1988).

² Kan här tolkas som användbarhet, tillgänglighet.

³ En liknande indelning har Reynolds (1992:339).

Dessutom skall verktyget innehålla en datakatalog samt säkerhetsregler. Att kalla ett kalkylprogram för ett fjärde generationens verktyg är enligt ovanstående definition något tveksamt, då datakatalogen ofta är rudimentär. Å andra sidan kan hävdas att kalkylprogram kan uppfattas som mer tillgängliga för utvecklare än många s.k. fjärde generationens verktyg (Holmberg 1995:53, Greve & Karlsson 1994:205).

15.2 Kalkylprogram

Kalkylprogram är en vanlig form av standardprogram. I litteratur där anveckling beskrivs eller relateras på annat sätt är det vanligt att kalkylprogram nämns som ett exempel på anvecklingsverktyg.

Världens första kalkylprogram VisiCalc, konstruerat av den amerikanske programmeraren Daniel Bricklin, kom kring 1980 och bidrog genom sin lättillgänglighet till spridningen av persondatorn (Nationalencyklopedin). Användning av kalkylprogram innehåller mycket av det som karakteriserar anveckling, nämligen tillgänglighet och användbarhet. Dagens kalkylprogram är långt mer avancerade än Bricklins VisiCalc, men den grundläggande principen är den samma. Kalkylprogrammet bygger på metaforen "rutat papper". Programmet är ursprungligen avsett för interaktiva beräkningar. Kalkylprogrammets centrum är det så kallade kalkylbladet, som består av rader och kolumner, och vars skärningspunkter kallas celler. I cellerna kan man mata in text, tal och formler. Formulerna kan bestå av konstanta värden och/eller referenser till andra celler. Formulerna kan också anta formen av funktioner av olika karaktär och komplexitet. Då nya värden matas in i cellerna räknas formulerna om, s.k. "what-if" analys. Kalkylprogram är vanligt förekommande på arbetsplatser. Enligt Isakovitz m.fl. (1995:1) har kalkylprogram blivit den vanligaste verktyget för beslutsstöd inom modern företagsekonomi. Tillämpningar gjorda med kalkylprogram (kalkylsystem) kan variera i utseende och komplexitet från mycket enkla, gjorda på några minuter, till komplicerade fleranvändarsystem med en utvecklingstid på flera år (Avdic 1995a:119ff).

Kalkylprogram betraktades ursprungligen inte som ett systemutvecklingsverktyg i traditionell bemärkelse (som t.ex. procedurella programspråk), även om det förekommer att system utvecklas traditionellt med hjälp av kalkylprogram. Andra programvarutyper av liknande karaktär är layout-, CAD- och statistikprogram. Likheten ligger i den verktygslänkande funktion som dessa programvaror har i och med att de kan utnyttjas för att skapa nya verktyg (i en annan praktik) eller för att utöka funktionaliteten (i sin egen praktik) i form av egna funktioner, mallar etc.. På programvarumarknaden finns idag ett antal väletablerade kalkylprogram⁴. Kalkylprogram bygger på samma grundläggande princip och kan därför sägas vara relativt likartade till sin karaktär.

⁴ Nämnas kan MS Excel, Quattro Pro och Lotus 1-2-3.

Tabell 21 Jämförelse: Langefors definition av IS och Pankos definition av kalkylprogram.

Langefors		Panko
Inmatning	↔	Kalkylblad(sanalys)
Lagring	↔	Databas (och kalkylblad)
Bearbetning	↔	Kalkylbladsanalys och makro
Presentation	↔	Grafik (och kalkylblad)
Överföring		

Panko (1988:388) delar in kalkylprogram enligt fyra grundläggande funktioner, nämligen kalkylbladsanalys (spreadsheet analysis), grafik, databas och makro. Denna indelning följer grovt de befintliga programfunktionerna och kan till viss del relateras till den klassiska definitionen av informationssystem som bestående av inmatning (kalkylblad(sanalys)), lagring (databas), bearbetning (kalkylbladsanalys och makro) och presentation (grafik).⁵ (Se Tabell 21 ovan)

Data i ett kalkylsystem lagras vanligtvis⁶ på ett *kalkylblad*. Ett kalkylblad kan ses som en sorts generell, komplex datastruktur, som tillåter olika former av datarepresentation. Kalkylbladet tillåter representation av strukturer, t.ex. matriser. Cellerna kan formateras på olika sätt, vilket är en väsentlig aspekt för anvecklare, som utnyttjar kalkylprogram för presentationsändamål, vilket för övrigt är en vanligt förekommande användningsform. (Avdic 1995a) Till själva kalkyldelen hör också ett behandlingsspråk med vars hjälp man kan utforma mer förfinade och/eller komplexa kalkylsystem än om man endast utnyttjade de direktmanipulerande delarna i själva kalkylbladet.

Sedan Bricklin skapade VisiCalc har en utveckling skett mot integration av alltfler funktioner i kalkylprogrammet, även om grunden i princip är densamma. 1999 kan ett kalkylprogram bestå av: (Avdic 1995a, Avdic 1997b)

- Möjlighet till grundläggande dokumenthantering t.ex. öppna, stänga, spara
- möjlighet att behandla data med formler
- tillgång till färdiga funktioner
- möjlighet att hantera matriser
- möjlighet till formatering av kalkylblad och diagram
- möjlighet att presentera data som diagram av olika slag, programmet kan innehålla ett hundratal färdiga diagramtyper
- möjlighet att skapa olika typer av rapporter
- möjlighet att anpassa användarens miljö i form av egendefinierade funktioner, egendefinierade format-, diagram- och dokumentmallar, egendefinierade sorteringsföljder, startkataloger samt anpassning av skärm- och tangentbordsfunktioner
- möjlighet till redigering av kalkylblad och diagram (även med s.k. ”drag and drop”)
- möjlighet att använda bilder, figurer och andra grafiska objekt
- möjlighet att använda textbehandlingsfunktioner

⁵ Överföringsfunktionen finns delvis implementerad i import- och exportfunktioner delvis finns den i den nätverksmiljö där användning av kalkylprogrammet äger rum.

⁶ Data kan också lagras i en extern databas.

- möjlighet att lagra metainformation av olika slag⁷
- tillgång till speciella funktioner för dataanalys
- möjlighet att lagra och behandla data i form av tabeller
- möjlighet att mata in data till tabeller via formulär
- möjlighet att referera till värden med beskrivande namn (utan att dessa explicit behöver deklarerats)
- möjlighet till säkerhet och skydd på olika nivåer (cell, blad och fil)
- möjlighet att granska den logiska strukturen i en kalkylmodell
- möjlighet att lagra kommentarer i anslutning till kalkylsystemet som text eller ljud
- gränssnitt mot externa databaser
- gränssnitt mot Internet
- funktioner för export och import av skilda filformat
- möjligheter till utskrift
- möjlighet att med dialogkomponenter, icke-procedurellt, skapa (som utvecklaren upplever som) professionella gränssnitt till egna kalkylmodeller
- möjlighet till integration med andra miljöer, t.ex. ordbehandlingsprogram
- möjlighet till icke-procedurell makroprogrammering (med inspelningsfunktion)
- möjlighet till makroprogrammering via ett programmeringsspråk
- tillgång till programmeringshjälpmedel t.ex. stegning, variabelbevakning, interaktiv hjälp med format och tillåtna egenskaper och metoder
- tillgång till guider för stöd vid genomförande av kommandon som kräver flera steg⁸
- tillgång till genvägar, snabbkommandon och faciliteter av olika slag, t.ex. transponering, seriegenerering, fönsterdelning och fönsterläsning,

Då avhandlingen syftar till att relatera olika kunskapstyper, KPA och kalkylsystem eftersträvas här en indelning av kalkylprogramfunktioner som är relaterad till KPA och det speciella med denna typ av aktivitet. KPA präglas av *integration* av systemutvecklingsaktiviteter, dvs analys, utformning, realisering, förvaltning, drift och användning. Denna integration för med sig ett dominerande inslag av *interaktivitet* i och med att datasystemets delar kan integreras på ett kalkylblad, vilket i sin tur ger möjlighet att snabbt växla mellan rollen som användare av systemet och som utvecklare av detsamma (se även sid 190 och 219). En kategorisering av kalkylprogrammets funktioner utifrån det interaktiva inslaget i KPA skulle kunna formuleras enligt följande:

- **Bearbetning** – formler, funktioner, matrishertering, makro
- **Relatering** – värden/formler, referenser/namn, länkning⁹
- **Analys (utan bearbetning)** – disposition, tabellhantering, dataanalys
- **Presentation** – formatering, utskrift, rapporter, diagram, grafik, exportfunktioner
- **Indata** – tabell, formulär, importfunktioner, gränssnitt mot externa databaser, gränssnitt mot Internet
- **Operatörsgränssnitt** – säkerhet, dialog, formatering

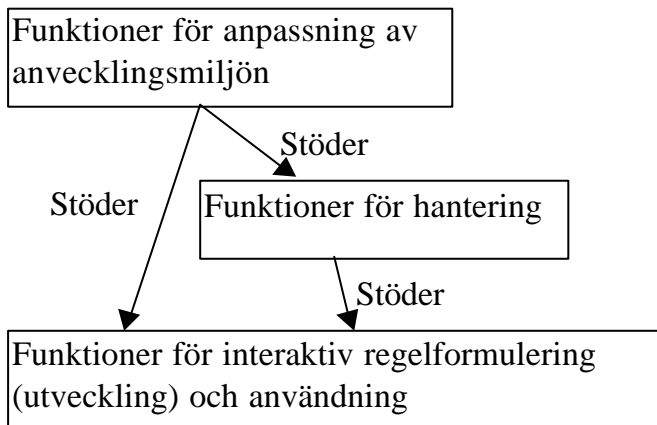
⁷ T.ex. listor med namnvariabler som används i kalkylsystemet och kommentarer som kan lagras i systemet.

⁸ T.ex. diagramkonstruktion, formelkonstruktion, funktionsformulering, etc.

⁹ Relatering skulle till viss del kunna ses som en aspekt av bearbetning, men också som indata eller presentation. Av det skälet har kategorin fått en egen rubrik.

- **Hantering** – grundläggande dokumenthantering, redigering, formatering¹⁰, guider, genvägar & specialfaciliteter
- **Applikationsutformning** – granskning, makro, programmeringshjälpmedel, integration av miljöer, metadatahantering
- **Förvaltning** – Grafik, textbehandling¹¹, metainformation, namn, granskning, hantering
- **Anvecklingsmiljö** – kalkylmiljöanpassning

Ovanstående indelning kan ses som en uppräknig av det interaktiva anvecklingsarbetets olika delar. Klassificeringen skiljer sig något från Pankos. *Bearbetning, relating, analys, presentation* och *indata* tenderar till att flyta samman till en integrerad helhet vilket underlättar interaktivitet vad gäller utveckling och användning av kalkylsystem. Dessa funktioner bidrar till att bearbeta informationsaspekten snarare än dataaspekten hos kalkylsystemet. Funktioner för utformning av *operatörsgränssnitt* kan användas om det finns en annan operatör av kalkylsystemet än anvecklaren själv. Möjligheterna till *hantering* har en väsentlig plats i det interaktiva utformningsarbetet, men på ett annat plan än de redan nämnda funktionerna. Dessa funktioner bidrar till att bearbeta dataaspekten snarare än informationsaspekten hos kalkylsystemet (i motsats till den första gruppen av funktioner). Hanteringsfunktionerna har till uppgift att stödja den första gruppen av funktioner. Funktionerna för *applikationsutformning* och *förvaltning* hör till utvecklardelen av anvecklarrollen. Även dessa kan användas interaktivt för informationshantering som den första funktionsgruppen. På den översta nivån finns funktioner för *anvecklingsmiljön*, som används för att anpassa miljön för användning och utveckling på olika sätt. Förhållandet mellan nivåerna visualiseras i Figur 104 nedan.



Figur 104 Funktioner i en interaktiv KPA-miljö.

I kapitel 10 (sid 195) visades att faktorer som berättigade anveckling var

- föränderlig verksamhet
- föränderliga informationsbehov
- krav på informationens aktualitet

¹⁰ Formatering kan dels användas för presentationsändamål, men även för hantering av systemet i övrigt. Genom att t.ex. ge vissa delar av ett kalkylblad viss färg kan dess status demonstreras. Formatering kan även användas för att reglera inregistrering till ett kalkylsystem, dvs för att skapa operatörsgränssnittet.

¹¹ Kan användas för dokumentation.

- krav på informationens kvalitet.

En föränderlig verksamhet innebär att verksamhetens struktur kan komma att förändras på något sätt. En strukturförändring kan t.ex. innebära nya produkter, ny organisation. Nya informationsbehov uppstår högst sannolikt i föränderliga verksamheter. En föränderlig verksamhet kan medföra att informationen måste vara aktuell. Att verksamheten strävar efter hög kvalitet på information behöver kanske inte motiveras. Om detta skall återspeglas i verksamhetens informationssystem kan det innebära

- att nya informationssystem måste utvecklas
- att nya typer av informationssystem måste utvecklas
- att befintliga informationssystem måste ändras
- att ny typ av information måste produceras
- att förändringarna måste genomföras utan dröjsmål
- att verifiering av informationen måste kunna ske fortlöpande

Om dessa krav skall mötas med KPA krävs att utvecklaren skall kunna utveckla och förändra kalkylsystem kontinuerligt. Detta kan underlättas av möjligheter till omedelbar parallell utveckling och användning samt till verifiering av informationens kvalitet. Kalkylprogrammets interaktiva, integrerade och generella karaktär kan av utvecklare uppfattas som överensstämmande med dessa krav.

15.3 Kalkylprogram och risker

Kalkylprogram är gjorda för att ge möjlighet att bygga, modifiera och utvidga kalkylsystem utan stora verktygskunskaper. Eftersom kalkylprogram är på detta sätt öppna (programmerbara och standardiserbara) till sin karaktär och speciellt konstruerade för interaktiv användning av verksamhetsföreträdare, finns det möjligheter att skapa kalkylsystem som innehåller fel och därmed risker av olika slag. Ledell (1993:48) påpekar att säkerhet inte är något som varit starkt förknippat med kalkylprogram.

Enligt Panko & Halverson Jr (1996) har man vid alla studier som gjorts angående fel i kalkylsystem hittat sådana fel. Felfrekvensen i kalkylsystem är ungefär som i TSU, men då utvecklingsförloppet är annorlunda sker inte felsökning på samma systematiska sätt i KPA som i TSU. Studier antyder även att felsökning i egna kalkylsystem kan vara ineffektivare än i kalkylsystem konstruerade av andra. Utifrån studier av 16 undersökningar, identifieras ett antal feltyper som kan uppstå vid utveckling (Panko & Halverson Jr 1996, Panko 1996).

Kvantitativa och kvalitativa fel

På en översta nivå skiljer man på kvantitativa och kvalitativa fel. *Kvantitativa fel* är numeriska fel som leder till felaktiga utdata. Exempel på detta kan vara en felaktig formel. *Kvalitativa fel* genererar inte omedelbara felaktiga utdata, utan innebär att fel kan uppstå då kalkylsystemet modifieras eller att felsökning försvåras, eller att förvaltning av systemet fördröjas eller att utdata kan misstolkas. Exempel på ett kvalitativt fel

kan vara ett kalkylsystem som innehåller konstanter i formler som måste ändras vid ny beräkning.

Mekaniska, logiska och utelämnande fel

Mekaniska fel är t.ex. skrivfel och referensfel. Enligt Panko & Halverson Jr är dessa fel relativt lätta att upptäcka. *Logiska fel* uppträder i form av felaktiga formler antingen p.g.a. fel algoritm eller p.g.a. fel formler utifrån rätt algoritm. Logiska fel kan underindelas i Eureka fel (lätta att styrka) och Cassandra fel (svåra att styrka). En annan indelning av logiska fel är i rena logiska fel och domänlogiska fel (beror på anvecklarens bristande verksamhetskunskap). *Utelämnande fel* innebär att något utelämnats från systemet. Dessa fel anses vara svåra att upptäcka (Panko & Halverson Jr 1996).

Osäkra indata

Ett tänkbart fel kan vara att anvecklaren kan mata in felaktiga eller osäkra data. Detta kan göras av olika skäl. I extremfallet kan en anvecklare med berätt mod mata in felaktiga data av något skäl. I andra fall kan indata eller systemutformning medvetet eller omedvetet anpassas till olika hänsynstaganden.

Strukturella fel

Kalkylprogrammet som sådant kan vara orsak till att vissa fel uppstår, och Panko & Halverson Jr (1996) anger vad man kallar *strukturella orsaker*. Till detta hör kryptiska formler och svårigheter att se mer än en formel i taget.

Ovannämnda fel kan i olika utsträckning förebyggas och/eller åtgärdas beroende på felets typ. Vissa fel beror i högre grad på anvecklarens kompetens och kunskaper än andra. Det kan finnas olika möjligheter att automatisera felgranskningen. Vissa fel kan endast förebyggas med domänkunskap. I kalkylprogram kan finnas stöd för undvikande av vissa fel. Detta stöd kan finnas i olika former. En variant är utvecklingsinteraktiv t.ex. situationsanpassad hjälp, felvärden och guider. En annan variant är ex post facto-orienterad t.ex. med hjälp av gransknings och verifieringsfunktioner. Exempel på denna funktionen *granskning* i MS Excel där relationer mellan formler visas grafiskt. En särskild typ av felförebyggande funktioner är funktioner i programmeringsspråket som ger möjlighet att stega program, bevaka variabler, eller att få hjälp med syntax vid programmering.

I Tabell 22, Tabell 23 och Tabell 24 nedan, struktureras de olika feltyperna i syfte att identifiera från vilket kunskapsområde kunskap kan utnyttjas för att förebygga eller åtgärda de olika felen.

Användare och utvecklare

Tabell 22 Kvantitativa och kvalitativa fel (med utgångspunkt från Panko & Halversson (1996)).

Feltyp	Orsak	Exempel	Fel förebyggs/åtgärdas med
Kvanti- tativa	Numeriska fel	Felaktiga formler	Användning av hjälpen Testning Granskning
Kvalitativa	Modifiering av kalkylsystem	Konstant i formel som måste förändras	Utformning av system - Strukturering - layout - Parameterstyrning Granskning

Tabell 23 Mekaniska, logiska och utelämnande fel.

Feltyp	Orsak/beskrivning	Exempel	Fel förebyggs/åtgärdas med
Mekaniska	Felaktig inmatning	Skrivfel Referensfel	Domänkunskap Användning av hjälpen Testning Granskning
Logiska	Fel formler p.g.a. - fel algoritm eller - feltolkad algoritm		
I a) Eureka	(Lätta att styrka)	Tidsverkningsgrad större än 100 %	Domänkunskap Testning
I b) Cassandra	(Svåra att styrka)	Komplexa formler i flera led	Domänkunskap Testning Granskning
II a) Rent logiska		Påslag minskar istället för ökar värde	Domänkunskap Testning
II b) Domänlogiska	(Bristande domänkunskap)	Felaktig beräkning av produktionsresultat	Domänkunskap Testning
Utelämnande	Något har utelämnats	En rabattsats saknas i offertberäkning	Domänkunskap Testning

Tabell 24 Osäkra indata och strukturella fel.

Feltyp	Orsak	Exempel	Fel förebyggs/åtgärdas med
Osäkra indata	Inmatning av osäkra eller felaktiga indata	Felaktigt produktionsresultat vid produktkalkyl	Domänkunskap
Strukturella	Kalkylprogrammets konstruktion	Kryptiska formler Svårt att se flera formler	Olika färger på referensdelar Granskning Visa formler i st för värden

Även om matchning av feltyper och kunskapsdomäner skulle kunna relateras på andra sätt, kan man ändå konstatera att det finns några huvudsakliga kunskapsdomäner för

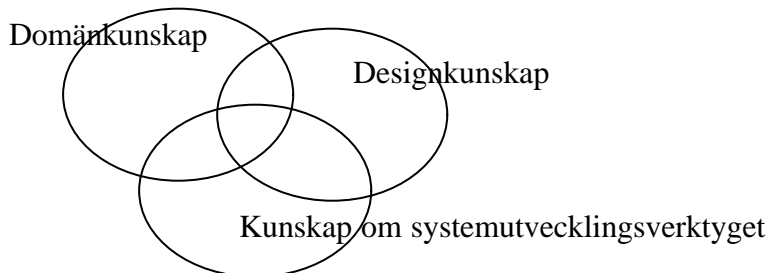
att förebygga och/eller åtgärda fel i kalkylsystem. De olika åtgärderna bygger på kunskap om

- Verksamheten (domänen)
- Utformning av kalkylsystem (design)
- Funktioner i kalkylprogrammet (utvecklingsverktyget)
- Testning (design, domän och verktyg)

När det gäller designaspekter kan dessa gälla såväl utformning av gränssnittet för operatören av informationssystemet som utformning av den information som systemet producerar. I de fall som det gäller anveckling av typerna *självförsörjning* eller *gruppanveckling* (se kap 11), där anvecklaren ingår i användargruppen, är inte fokus på operatörsgränssnittet lika viktigt som när det gäller *fadderanveckling* (eller TSU). Där emot spelar utformning av utinformationen en viktig roll ur ett pragmatiskt perspektiv eftersom anvecklaren i första hand utför arbetsuppgifter mot mål. Utinformationen bör utformas så att önskad effekt erhålls då den riktar sig till slutliga informationsmottagare utanför gruppen där anvecklaren verkar.

Olika typer av testning som kan göras är test med:

- Verkliga värden
- Enkla värden
- Rimliga värden
- Ytterlighetsvärden



Figur 105 Olika kunskapsområden, relevanta för förebyggande av fel i kalkylsystem.

De kunskapsområden som visas i Figur 105 ovan kan ses i relation till diskussionen i kapitel 1, där det framgår att de kunskapsområden som är väsentliga för anveckling är verksamhetskunskap, yrkeskunskap och kunskap om systemutvecklingsverktyget. Relaterat till Figur 105 ovan kan domänkunskap delas upp i verksamhetskunskap och professionskunskap, medan designkunskap och kunskap om utvecklingsverktyget tillsammans utgör kunskap om utvecklingsverktyget.

15.4 Användbarhet

Vid TSU fästes ofta stor vikt vid användbarhet. Man diskuterar t.ex. design av gränssnitt och applikationers funktionalitet. Vid TSU konstruerar IT-specialister system åt verksamhetsföreträdare. Eftersom systemen måste gå att använda är gränssnittet viktigt. I interaktiva tillämpningar är det rent av kritiskt. Ett bankomatsystem måste vara

enkelt och entydigt att använda. Ett realtids priskalkyleringssystem som skall användas av flera handläggare vid kontakt med kunder via telefon måste det vara enkelt att använda. Det utvecklade systemet måste fungera som ett verktyg för att uppfylla sitt syfte.

Användbarhetsbegreppet kan även relateras till anveckling. Dels kan applikationer oavsett vem de är utvecklade av sägas vara mer eller mindre användbara, dels så kan utvecklingsverktyget vara användbart i olika avseenden, samtidigt som det kan sägas ge möjlighet till olika grad av användbarhet i den färdiga applikationen. Det är följaktligen två verktygsgränssnitt som aktualiseras vid anveckling, nämligen kalkylprogrammets och kalkylsystemets. Anvecklaren använder ett verktyg (kalkylprogrammet) för att skapa ett annat verktyg (kalkylsystemet). Kalkylprogrammets gränssnitt är viktigt men till största delen opåverkligt. Kalkylsystemets gränssnitt är till stor del en följd av kalkylprogrammets men påverkligt på många sätt. Man kan välja hur systemet skall startas och avslutas, hur och var data skall matas in, bearbetas och matas ut. Man kan välja utformning av skärmbilder och utskrifter.

En fråga som kan ställas i anslutning till detta är i vilken utsträckning utvecklare är betjänta av att känna till teorier om användbarhet. En annan fråga är vad det är i utvecklingsverktyget som gör det mer eller mindre användbart och på så vis gör det mer tillgängligt för utvecklaren, vilket i sin tur kan hjälpa till att eliminera kunskapsöverföringsproblemet.

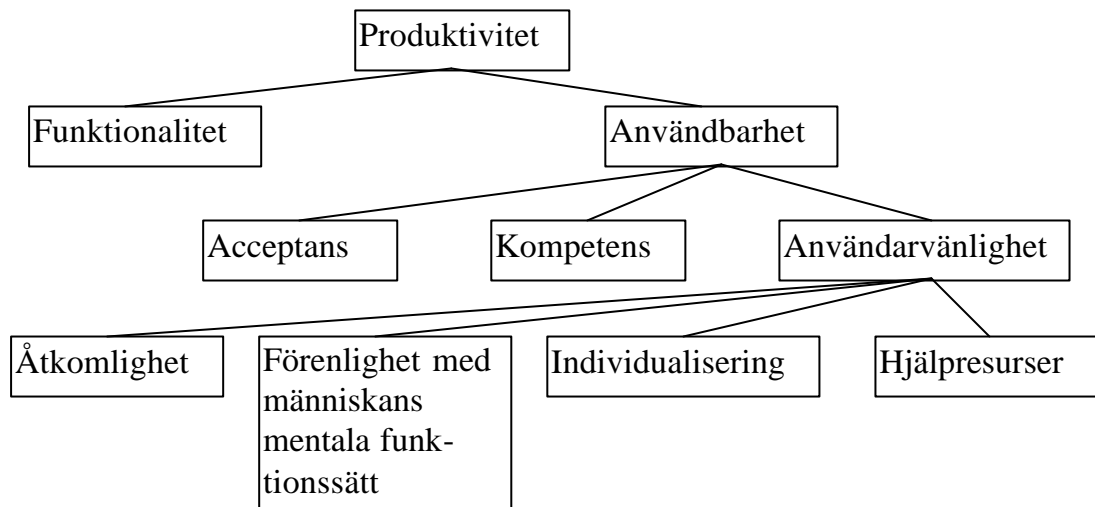
Användbarhet är en aspekt av anveckling som är förknippad med såväl utveckling som användning av kalkylsystemet. Kopplingen till utveckling består i att utvecklingsverktygets användbarhet är avgörande för om utvecklaren alls kommer att använda sig av det. Kopplingen till användning består i att det färdiga systemet helst skall vara användbart för att utnyttja.

Nielsen (1993) menar att användbarhet förknippas med följande fem egenskaper:

- **Lättlärdhet:** Systemet skall vara lätt att lära, så att användaren snabbt kan börja utföra sina arbetsuppgifter.
- **Effektivitet:** Systemet skall vara effektivt att använda, så att användaren kan nå en hög produktivitetsnivå när han/hon väl lärt sig använda det.
- **Lätt att minnas:** Systemet skall vara lätt att komma ihåg, så att den som använder systemet bara ibland, lätt kan komma tillbaks till systemet efter en längre tid utan att behöva lära sig det från början igen.
- **Felhantering:** Systemet ska leda till att dess användning har låg felfrekvens. Användaren ska därmed göra få fel och de fel han/hon gör ska lätt kunna åtgärdas. Fel med katastrofala följder får inte förekomma.
- **Tillfredsställelse:** Systemet skall vara angenämt att använda.

Enligt Allwood (1991:9f) kan produktiviteten (kvaliteten hos ett arbetsresultat) hos ett system eller ett program beskrivas i termer av användbarhet och funktionalitet. Med *funktionalitet* avses ett programs/systems möjlighet att klara relevanta arbetsuppgifter. Vidare ser Allwood *användbarhet* som bestående av användaracceptans, användar-

kompetens och användarvänlighet. Användarvänlighet består i sin tur av åtkomlighet, förenlighet med människans mentala funktionssätt, individualisering och hjälpresurser (se Figur 106 nedan).



Figur 106 Produktivitet enligt Allwood (1990:11).

Det finns även en ISO-standard för användbarhet, ISO-9241-11, som förknippar begreppet med effektivitet och tillfredsställelse. ISO-definitionen poängterar även att användbarhet endast kan ses relativt ett systems användare, vilket innebär att användbarheten varierar med person (Fossum 1996:3). Denna distinktion antyder att egenskaperna som räknas upp är mer eller mindre beroende av användaren av systemet. Olika personer kan ha olika lätt att lära ett givet system eller kan uppleva olika grad av tillfredsställelse vid användning av systemet.

Smithson & Hirschheim (1990) beskriver ett systems gränssnitt som en representation av en *mental modell* av problemområdet. Den som utformar ett system har en mental modell av användaren och problemområdet. På samma sätt ger ett gränssnitt en användare en mental modell av det problemområde som systemet relaterar till.

...the systems designer has a mental model of the user and the problem situation, and that he/she, consciously or unconsciously, employs that model in designing the system. This model becomes embedded in the system, its training procedures and documentation and thus may act as a significant input to the user's construction of his/her mental model. (Smithson & Hirschheim 1990:37ff)

Diskussionen om mentala modeller skulle kunna tolkas som att anvecklade system i denna mening är mer relevanta än traditionellt utvecklade eftersom utvecklarens mentala modell av problemsituationen är samma som användarens, det är ju samma person. Om utvecklaren anvecklar för en medarbetare är möjligheten att man till stor del delar den mentala modellen icke försumbar, då man verkar inom samma verksamhet.

Definitionerna av användbarhet ovan bygger på en oklar definition av informationssystem. Underförstått kan det tolkas som att de system som nämns i definitionerna är informationssystem utvecklade av IT-specialister för användare. Användbarhet dryftas i avsikt att utveckla kunskaper om design av system att användas av utvecklare vid

systemutveckling. I vid mening kan både kalkylprogram och kalkylsystem betraktas som informationssystem. Några skillnader kan dock noteras.

Ett *kalkylprogram* är ett ytterst generellt informationssystem (se definitioner av kalkylprogram ovan), som inte är skapat för en avgränsad uppgift utan snarare för att klara många uppgifter. Olika delar av ett kalkylprogram kan därför vara olika användbara i olika sammanhang. Funktionalitetsegenskapen är mer *systemrelaterad* än andra och är inte variabel på samma sätt som t.ex. lätlärdhet, som kan, i viss utsträckning, förmodas bero på utvecklarens egenskaper, t.ex. kompetens. Alla har t.ex. inte lika lätt att lära sig att använda kalkylprogram. Ett exempel på systemrelaterade egenskaper är att ett kalkylprogram kan innehålla funktioner för databasåtkomst oavsett vilken användare som använder systemet. På samma sätt kan man säga att funktioner för felsäkerhet antingen finns eller inte finns oberoende av användare. En användbarhetsegenskap för kalkylprogram kan alltså vara riklig funktionalitet och att denna skall vara lätt att nå när den behövs (åtkomlighet).

Ett informationssystem är definitionsmässigt anpassat för ett visst ändamål (se sid 161). Avsikten är normalt, när det gäller TSU, att det levereras i färdigt skick. Ett *kalkylsystem* skiljer sig från ett TIS genom att det normalt förutsätts att det skall kunna förändras av den som använder systemet (en följd av integration av utveckling, användning och förvaltning), vilket inte gäller för TIS. Det faktum att kalkylsystem används av utvecklaren innebär att *acceptansaspekten* inte är likvärdig för de båda typerna. Eftersom utvecklarens verksamhet är integrerad och interaktiv tillkommer användbarhetskrav som har att göra med själva utvecklingsverksamheten. Exempel på tillkommande användbarhetskrav skulle kunna vara *förändringsbarhet* och *komplexitetshantering*. Detta behandlas vidare i nästa avsnitt.

Utifrån resonemanget om att användbarhetsegenskaper skulle kunna vara mer eller mindre systemrelaterade respektive mer eller mindre aktörsrelaterade identifieras följande användbarhetsfaktorer:

Mer systemberoende egenskaper:

Funktionalitet; Funktionalitet (Allwood)

Effektivitet; Effektivitet (Nielsen), Åtkomlighet (Allwood)

Lärbarhet; Lätlärdhet + Lätt att minnas (Nielsen), Förenlighet med människans mentala funktionssätt + Individualisering (Allwood)

Felhantering; Felhantering (Nielsen), Hjälpresurser (Allwood)

Mer aktörsberoende egenskaper:

Acceptans; Acceptans (Allwood), Tillfredsställelse (Nielsen)

Kompetens; Kompetens (Allwood)

Faktorerna inom respektive grupp kan till viss del överlappa varandra. Mellan grupperna råder vissa samband. Ett exempel på sådant samband är Funktionalitet-Acceptans. Huruvida ett system är funktionellt beror på verksamhetens behov. Skall

sedan systemet användas måste användaren uppleva ett visst mått av acceptans. Alla systemberoende egenskaper kan relateras till användarens/anvecklarens Kompetens.

Då utvecklaren även är användare förefaller inte Lärbarhets- och Acceptansfaktorerna ha samma roll som vid TSU. Ett undantag är när fler än utvecklaren själv skall använda systemet. De användbarhetsfaktorer som skulle kunna vara särskilt intressanta vid anveckling är följaktligen

- Funktionalitet
- Effektivitet
- Felhantering
- Anvecklarens kompetens

15.5 Utvecklingsverktygs användbarhet

Systemutvecklingsverktyg kan indelas i grupper efter hur maskinnära de är, dvs hur mycket funktionalitet som finns färdigt i programvaran från början. Anvecklarverktyg skiljer sig från mer maskinnära verktyg bl.a. genom högre grad av tillgänglighet av funktionalitet. En speciell typ av utvecklingsverktyg är generella standardprogram av typen databashanteringsprogram, CAD-program, statistikprogram och kalkylprogram. Det som utmärker dessa program är att de är möjliga att använda med liten utvecklingsinsats men att de kan utnyttjas för mer omfattande utvecklingsarbete också. Denna typ av program innebär att en stor del av de potentiella informationssystemen är skapade i förväg av programvaruleverantören, som på detta sätt kan sägas ha formulerat en hypotes om vilka generella funktioner som framtida anvecklare/utvecklare kan tänkas ha nytta av, och försett utvecklingsverktyget med en del av dessa.

Det är en flytande gräns mellan systemutvecklingsverktyg och informationssystem på samma sätt som det är det mellan utveckling och användning (se kap 1.2.1). Ett kalkylprogram kan ses både som ett utvecklingsverktyg och ett IS. Kalkylprogram ingår som en del i kalkylsystem samtidigt som kalkylprogram är utvecklingsverktyg för anvecklare. Detta innebär att användbarhet för kalkylprogram förväntas gälla både för kalkylprogrammet som utvecklingsverktyg och som informations(kalkyl)system för användning.

"In many applications (e.g. word processing, desktop publishing, spreadsheets and hypertext) current software packages are highly sophisticated and complex, such that the distinction between 'programming' end-users and those that merely operate the packages becomes increasingly meaningless. ... Rather than a particular system being developed and then operated, as in the traditional life cycle model, the situation becomes more one of end-users acquiring sophisticated tools and gradually learning to personalize and tailor the packages to their needs, and then to use the packages for more complex activities." (Smithson & Hirschheim 1990:37ff)

Under de år som datorer funnits har det skett en utveckling mot att programspråk tagit hand om mer och mer av automatiserbara rutiner. Mer och mer av det som systemutvecklaren själv fått sköta när han använt t.ex. assemblerspråk har programmerats in i

utvecklingsverktyg, vilket har lett till kortare utvecklingstider eftersom rutiner finns färdiga att utnyttja för t.ex. bildskärmshantering. En annan fördel med färdiga rutiner utöver tidsbesparing är att de kan vara mer felsäkra i och med att de är testade. Den förenklade programmeringen har även vissa nackdelar, den kan t.ex. leda till mindre ändamålsenliga system då de färdiga rutinerna kanske inte passar i aktuell verksamhet. Om t.ex. bildskärmshanteringen inte täcker de behov som finns i verksamheten så kan icke önskade effekter uppstå. Å andra sidan skulle systemutveckling i ett maskinnära språk som assembler kunna bli så tidsödande att det aldrig övervägs överhuvudtaget. När det gäller den flexibilitet som ovan förknippats med maskinnära språk, så kan hävdas att en annan typ av flexibilitet kan uppstå genom att det finns tillgängligt ett antal färdiga funktioner som kan användas på olika sätt. Ett exempel på detta skulle kunna vara en diagramfunktion med vars hjälp olika sätt att representera data grafiskt skulle kunna genomföras. Mot bakgrund av ovanstående resonemang kan några användbarhetsfaktorer identifieras som sammanhänger med olika systemutvecklingsverktyg.

Funktionalitets befintlighet

Funktionalitets befintlighet ger flexibilitet att anpassa ett kalkylsystem till olika önskemål, vilket skulle kunna benämnas *förändringsbarhet*. Det kan gälla speciella önskemål om speciella funktioner iden aktuella utvecklingsprogramvaran, t.ex. att kunna representera utdata med viss diagramtyp. Det kan även gälla att utifrån verksamhetsbehov kunna anpassa kalkylsystemet till hårdvara (in- eller utmedier) t.ex. kvittoskrivare, läspenna etc. Det kan även gälla anpassning till programvarumiljöer, t.ex. PC till MAC eller PC till stordator. Till utvecklingsverktygets funktionalitet kan även räknas komplexitetshantering, som innebär att komplexa behov i verksamheten skall kunna formaliseras i kalkylsystemet. Funktionalitets befintlighet är en grundläggande faktor för anveckling.

Funktionalitets tillgänglighet

Funktionalitets tillgänglighet handlar om möjligheten att kunna utnyttja befintliga funktioner. Hög tillgänglighet ger flexibilitet att tillgodose olika verksamhetsbehov och att pröva olika lösningar. Denna *flexibilitet* kan vara lämplig om verksamheten har olika typer av behov, om verksamheten är föränderlig och om informationsbehovet inte är explicit formulerat (t.ex. beslutsstödssystem), vilket innebär

- att programvaran under utvecklingsarbetet skall tillhandahålla interaktiv funktionalitet
 - att det färdiga informationssystemet skall tillhandahålla interaktiv funktionalitet.
- Relaterat till de faktorer som berättigar anveckling (se kap 10.3.4) är funktionalitets tillgänglighet en faktor som är relaterad till dessa. Funktionalitets tillgänglighet betyder att utvecklingsverktyget skall vara praktiskt användbart för utvecklare, dvs kunna användas av personer som ej är IT-specialister. Andra exempel på funktionalitets tillgänglighet är tillgång på maskin- och programvara, hårdvaruprestanda och programvaruprestanda.

Felhantering/Felsäkerhet

Ett programutvecklingsverktyg kan vara utrustat med funktionalitet för att hindra att användare¹² utför icke tillåtna åtgärder. Vid utveckling av TIS, kan arbete med att skapa denna typ av felsäkerhet ta mycket stor tid i anspråk. Utvecklingsarbetet (vid anveckling) är en balansgång mellan felsäkerhet/ tidsåtgång/ verktygskunnande/ dokumentationsbehov/ fleranvändaraspekter/tillgång till befintlig funktionalitet.

Utvecklingsverktygskunskap

Ju större kunskap om utvecklingsverktyget desto större tillgång till befintlig funktionalitet, vilket innebär att denna faktor kan ersätta funktionalitets tillgänglighet och skapa flexibilitet. Det ger även större möjlighet till felsäkerhet och minskar tidsåtgång. T.ex. berättade Ismo om hur kunskap om hur man utnyttjar cellskydd underlättade hans arbete.

Tidsåtgång

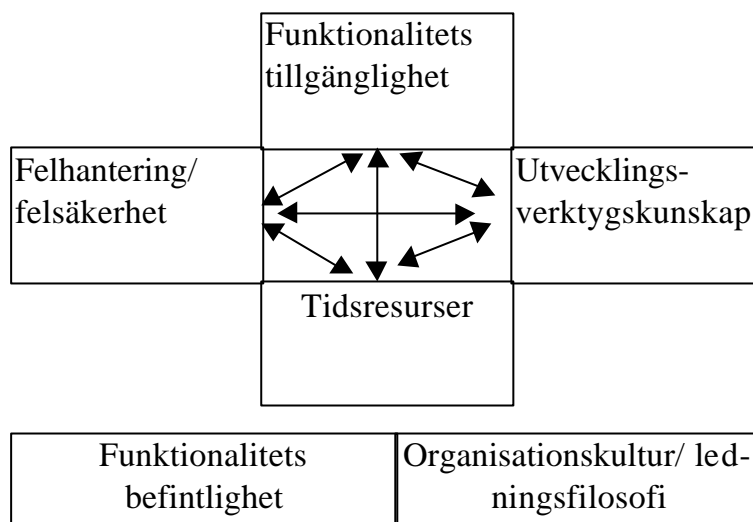
Om kraven på felsäkerhet inte är så stora, sparas tidsresurser vid utvecklingsarbetet. Funktionalitetens befintlighet och tillgänglighet samt utvecklingsverktygskunskap sparar också tidsresurser.

Organisationskultur/ledningsfilosofi

Ytterligare en faktor som kan spela in är den organisationskultur/ledningsfilosofi som indirekt styr huruvida och på vilket sätt anveckling kan äga rum i en verksamhet (se sid 192 och Avdic 1995a:78;97;113)). Detta är en grundläggande användbarhetsfaktor för anveckling.

Av de nämnda faktorerna är den första och den sista (funktionalitets befintlighet och organisationskultur/ledningsfilosofi) grundläggande och styr de övriga fyra. Övriga fyra faktorer är relaterade till varandra på så vis att funktionalitets tillgänglighet kan minska behov av kompetens, tidsåtgång och felhantering. Kompetens kan minska behov av funktionalitets tillgänglighet, felhantering och tidsresurser osv. (se Figur 107 nedan). Detta resonemang innebär att ju mer tillgänglig funktionalitet som byggs in i utvecklingsverktyg (t.ex. kalkylprogram) desto mindre utvecklingsverktygskunskap krävs för att utveckla system och desto fler kan dra nytta av sin verksamhetskunskap för att utveckla a-system som motsvarar verksamhetens föränderliga behov.

¹² Av utvecklingsverktyget, dvs i egenskap av utvecklare.



Figur 107 Användbarhetsfaktorer för anveckling.

15.6 Cyklisk utveckling av verktyg¹³

Utvecklingen från TSU mot KPA kan liknas vid den process genom vilken skriftspråket blivit tillgängligt för människor. Förr i tiden var läs- och skrivkunnighet förbehållet ett fåtal. Då en person utan dessa kunskaper hade behov av att få något skrivet eller läst tvingades han vända sig till en person med dessa kunskaper. Den hjälpsökande var utlämnad till specialistens kompetens och goda vilja. Genom allmän folkskola och ökad tillgång på böcker och papper blev det skrivna språket var mans egendom, åtminstone i Sverige. Med kunskap och materiella medel blev specialisternas kunskaper allmän egendom. Ökad skriv- och läskunnighet förändrade tillvaron för människorna, t.ex. vad gäller möjligheter till demokrati och inflytande enligt devisen "Kunskap är makt". Kunskapen kunde erhållas genom läsning. Men trots läs- och skrivkunnighet fanns det utrymme för specialister inom området. Hela yrkesgrupper, t.ex. sekreterare och grafiker, har livnärt sig på att tillhandahålla skriv och redigeringstjänster. Faciliteter som skrivmaskinen, datorn, ordbehandlingsprogram, layoutprogram etc. har återigen gett tillbaka möjligheten till icke-specialister att utföra det som tidigare betraktades som specialisternas förbehåll och privilegium. Många fördomar om experters oumbärlighet har rensats bort på vägen. Folkskolan var dock inte en reform som sågs med blida ögon överallt. Alla uppskattade inte konsekvenserna av att läskunnigheten ökade. Detta kunde t.ex. gälla de som själva kunde läsa och kunde dra nytta av det. Andra fördomar av mindre allvarligt slag har fått stryka på foten tack vare nya grundförutsättningar. Skrivmaskinskurser har t.ex. inte samma funktion idag som för tjugo år sedan och inte är det så noga hur man håller fingrarna då man skriver på ordbehandlare. Skriver man fel kan det ju enkelt rättas till.

Tillgången till kalkylprogram och därmed möjligheten till systemutveckling för utvecklare skulle kunna jämföras med tillgång till skriftspråket och därmed möjligheten att oberoende av specialister tillgodose upplevda behov. KPA skulle kunna liknas vid

¹³ Huvudsakligen hämtat från Avdic (1995a:215ff).

ett "informationsbehandlings självhushåll". Detta självhushåll innebär en utveckling från specialisering till generalisering/integration, dvs mot en situation där användare kan tillgodose sina informationsbehov med mindre och mindre hjälp av IT-specialister.

En jämförelse med framväxten av mänsklig civiliserad kultur kan tjäna som liknelse. Enligt Nissen (1988) är följande fenomen typiska kännetecken på att en civiliserad kultur etablerats:

- specialisering av hantverk, t.ex. krukmakare (Nissen 1988:28ff)
- administration, t.ex. viss beskattning (Nissen 1988:83)
- gemensam lag, rättsnorm, t.ex. att alla skall betala skatt och brottsanpassad straffnivå (öga för öga, tand för tand) (Nissen 1988:60f)
- monumentalskulpturer och arkitektur, t.ex. Zigguraten i Ur¹⁴ (ett tempeltorn) (Nissen 1988:98ff)

Mellan de ingående komponenterna råder ett dialektiskt förhållande. Arkitektur förutsätter specialisering och administration (som i och för sig också är en form av specialisering). Komponenterna förutsätter också ett område av homogen utvecklingskaraktär. Detta område har yttre och inre gränser: Området måste vara så stort (yttre gräns) så att invånarna kan få ett överskott. Området måste vara så litet (inre gräns) så att alla kan ha kontakt med det administrativa området, det urbana centret.

Verktygsutvecklingen i en civiliserad kultur utvecklas i cykler, som växelvis innebär specialisering respektive generalisering/integration. Skriftspråket har redan nämnts, men det finns fler exempel. Krukmakarhantverket kan tjäna som exempel. Till att börja med kunde krukmakaren ensam tillverka krukor. Så småningom utvecklades verktyg, t.ex. drejskiva och ugn, som gjorde det möjligt för andra än krukmakaren att tillverka krukor. Vår vardag är fylld av mer eller mindre komplexa bruksföremål som under historiens gång i cykler utvecklats mellan specialister och användare, mellan specialisering och generalisering. För att ta några exempel på verktyg som gett dess användare möjlighet att utföra specialisters arbetsuppgifter: kamera, skruvmejsel, rakapparat, stickmaskin, skrivmaskin, spis, telefon.

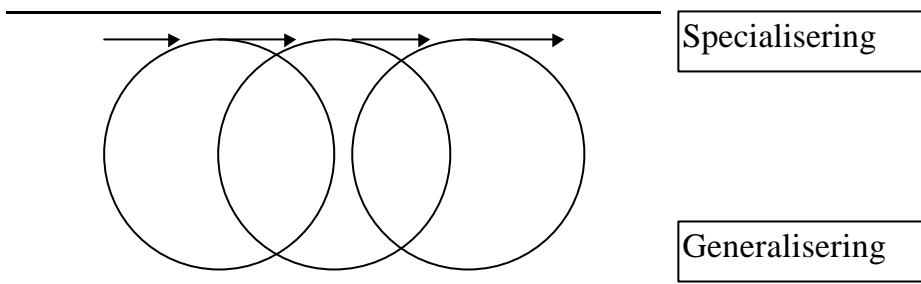
IT-områdets motsvarighet till ovan beskrivna utveckling för civiliserade kulturer skulle kunna vara följande:

Specialisering av hantverk

Informationsbehandling har alltid funnits, muntlig som skriftlig. När datorerna på 40- och 50-talet introducerades, krävdes specialkunskaper för att hantera dessa. Endast några få personer (hantverkarna/IT-specialisterna) hade kunskaper, som gjorde det möjligt att utforma informationssystem. Sakta men säkert har utvecklingen gjort det möjligt för allt fler att självständigt kunna utnyttja datorer för att tillgodose informationsbehov. På samma sätt som det funnits cykler för verktygsanvändning i historien har utvecklingen inom IT-området nu nått ett visst mått av generalisering (se Figur 108 nedan). Generaliseringen ligger i det faktum att personer utan specifika IT-specialistkunskaper kan tillgodose informationsbehov av en komplexitet som tidigare

¹⁴ Monumentalskulpturer och arkitektur fyller funktionen att manifesteras en civilisations makt och bibehålla värnaden för gudar och konungar.

inte varit möjlig. Ett uttryck för denna utveckling mot generalisering av verktyg och integration av uppgifter (användning och utveckling) är KPA.



Figur 108 Beskrivning av den cykliska utvecklingen av verktyg

Administration

Specialiseringen i form av administration går också in i en generaliseringsfas, så tillvida att de organisationer som krävts för att administrera specialisthanteringen (datacentralerna) nu inte fyller samma funktion. Dataavdelningar spelar en mer perifer roll vad gäller KPA än TSU. Dataavdelningarnas relation till anvecklarna är annorlunda då anvecklarna utvecklar system själva än då dataavdelningarna gör det åt dem. Anvecklarnas roll präglas vid KPA av större självständighet än vad användarnas gör vid TSU.

Gemensam lag, rättsnorm

Den gemensamma ”lag”, som präglat det hittills dominerande systemutvecklingsparadigmet är Livscykelmodellen.¹⁵ Som diskuterats ovan bryter KPA mot Livscykelmodellen på avgörande punkter, vilket innebär att den utveckling mot generalisering som beskrivits ovan även innebär en utveckling mot en ny rättsnorm och mot ett nytt paradigm.

Monumentalskulpturer och arkitektur

Att spekulera i motsvarigheter inom ämnesområdet IT på denna punkt kan måhända tyckas djärvt, men nog kan det betraktas som ganska klart att datorer uppfattas som komplicerade och svårhanterliga av personer med begränsade kunskaper om datorer och deras användning. För användare på företag med stordatorer kan emellanåt dataavdelningar framstå som uttryck för en viss form av ouppnåelig makt även om de är aldrig så användarvänliga. Bilder i böcker och tidningar visar datorhallar med slutna och komplexa system. Programmerare och andra IT-specialister kan lite tillspetsat ses som ett slags datakulturens överstepräster.

Om hantverksutvecklingen i IT-kulturen går in i sin integrerade fas av cykeln skulle detta utifrån liknelsen med civilisationsutvecklingen leda till förändringar även vad gäller administration, rättsnormer och arkitektur. I vilken riktning detta kommer att ske och vilka effekter det kommer att få behandlas implicit i denna avhandling.

¹⁵ En lite tillspetsad formulering.

15.7 Sammanfattning - Utvecklingsverktyg

Med verktyg avses i detta sammanhang hjälpmedel för kalkylprogramanveckling. Verktyg vid kalkylprogramanveckling kan vara kalkylprogram, annan teknik samt metoder och modeller.

Gränserna mellan kunskap, normer och verktyg är inte entydiga. Verktyg i form av metoder och modeller kan även relateras till områdena kunskap och normer i den praktikgeneriska modellen.

Kalkylprogram kan räknas till programmerbara, standardiserbara standardsystem. Andra verktyg av liknande typ är databashanteringsprogram och desktopprogram. Kalkylprogram är interaktivt, integrerat, generellt och innehåller (a) funktioner för anpassning av anvecklingsmiljön, (b) funktioner för hantering samt (c) funktioner för interaktiv regelformulering (utveckling) och användning. Kalkylprogrammets egenskaper underlättar för anvecklaren att utveckla och förändra kalkylsystem kontinuerligt, vilket behövs för att motsvara verksamhetens föränderlighet och krav på informationskvalitet.

Kalkylprogrammets egenskaper innebär risker, vilka kan leda till fel av olika slag. Felen kan förebyggas med kunskap om verksamhet, professionen samt om utvecklingsverktyg.

Användbarhetsbegreppet kan ses i relation både till utvecklingsverktyget och till det utvecklade systemet. Grundläggande användbarhetsfaktorer som är intressanta vid anveckling är funktionalitets befintlighet och organisationskultur/ledningsfilosofi. Utöver dessa faktorer är följande faktorer relevanta vid anveckling: funktionalitets tillgänglighet, felsäkerhet/felhantering, utvecklingsverktygskunskap och tidsresurser.

Verktyg genomgår i samhället en cyklisk utveckling, som skiftar mellan generalisering och specialisering. I kalkylprogram finns formaliserad och generaliserad kunskap tillgänglig för en större grupp personer än vad som var fallet innan kunskapen implementerades i kalkylprogram.

DEL IV

Avslutning

"Anar du gäddans
moltysta vinterparlament
där ute under dov-isen?"
(Ulf Eriksson 1987)

- *Syftet med detta kapitel är att relatera empiri och teori för att fördjupa kunskaperna om kalkylprogramanveckling.*

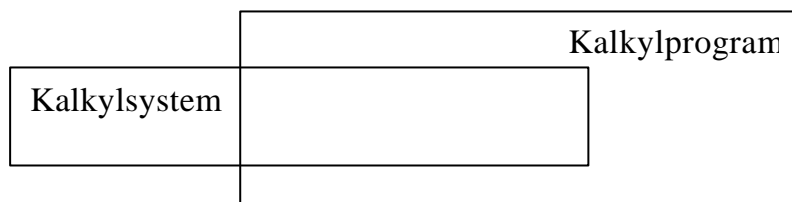
16 Analys av anvecklingsbegreppet

I *Del II, Empiri* och *Del III, Referensram* har fenomenet kalkylprogramanveckling analyserats fortlöpande. I slutet av respektive empirikapitel (kap 3-6) sammanfattas resultaten från de fyra studierna. Resultaten från analyserna i teorikapitlen (kap 7-15) finns även de sammanfattade i slutet av respektive kapitel. I detta kapitel utförs analys som har sin grund i såväl empiri- som teoridelarna. Då teoridelarna i sig själva har ambitionen att innehålla analys av de olika aspekterna av anveckling kommer delar av detta inte att återupprepas i detta analyskapitel. Redovisningen i detta kapitel följer den modifierade praktikgeneriska modellen och följer i stort sett samma ordning som i *Del III*. Vissa analyserade fenomen kan inte enkelt placeras under enstaka del i modellen och har därför placerats där det ansetts närmast höra hemma. De praktikgeneriska modulerna Verktyg och Normer behandlas i kapitel 17, som kan ses som relaterat till såväl kunskap, normer som anveckling.

16.1 Kalkylsystem

Kalkylsystem är informationssystem, fast inte av traditionell typ. Nurminens (1987) HIS-modell förklarar det karaktäristiska med kalkylsystem på ett träffande sett (se Tabell 8 sid 170). I kalkylsystemet ingår det verktyg som kalkylsystemet är byggt med (Avdic 1995:185). Se även Figur 109 nedan. I den licentiatavhandling (Avdic 1995a), som ligger till grund för denna avhandling, klassificerades kalkylsystem i fyra klasser: 1) Ordbehandlarsystemet, 2) Lilla kalkylsystemet, 3) Stora kalkylsystemet och 4) Applikationen (se kap 8.3). De i empiridelen relaterade kalkylsystemen är mestadels av typen Lilla och Stora kalkylsystemet. Av de intervjuade respondenterna hade endast

Olle på Myndigheten byggt kalkylsystem innehållande kod. Exempel på hur system med tiden byter systemklass genom att öka i komplexitet finns såväl på Myndigheten (**bokslutssystem**) som på Industrin (**uppföljning av arkningsmaskiner**). De första tre kalkylsystemtyperna kan ses som typiska kalkylsystem.



Figur 109 Delar av kalkylprogrammet ingår i kalkylsystemet.

I kapitel 6 redovisas ett exempel på ett kalkylsystem av typen Applikation. Kalkylsystem av denna typ kan ses som en form av användarprototyping. Skiljelinjen mellan a-system och traditionella system är inte alltid skarp men kan sägas gå där systemägaren är systemutvecklingsansvarig som i det relaterade fallet. I de fall som systemutvecklingsarbetet kräver specialkunskaper, som anvecklaren inte har, kan delar av utvecklingsarbetet utföras av IT-specialister. I dessa fall är väsentliga delar av analysarbetet redan utfört. Denna form av användarprototyping, kallas av Karlsson (1998) för *hybridmetod*¹. Den kan tillgripas medvetet, som i det fall som relateras i kapitel 6 eller så kan det vara ett sätt att hantera en utvecklingssituation där komplexitetsgraden i ett a-system blir anvecklaren övermäktig. Mika redovisade i kapitel 5 hur han fått ta över alltför komplexa kalkylsystem och strukturerat systemen på ett mer traditionellt sätt. Det utvecklade a-systemet kan utgöra prototyp för ett mer traditionellt utvecklingsarbete. Ovanstående resonemang kan ses som en antydning till lösning på de fall där komplexiteten i ett system blir anvecklaren övermäktig.

16.2 Anveckling

I avhandlingen har jämförelser gjorts med TSU enligt livscykelmodellen. Jämförelsen har sammanfattats i Tabell 12 sid 183. Anveckling kan enligt denna jämförelse ses som ett diametralt annorlunda sätt att bedriva systemutveckling. Kännetecknen för KPA redovisas i kapitel 10.2. Anveckling kan fungera som ett alternativ och komplement till TSU. Exempelen på kalkylsystem i Empiridelen skulle i de flesta fall inte kommit till om alternativet till anveckling varit TSU. Skälen är flera. Kalkylsystemen har i de flesta fall tillverkats inom ramen för det löpande arbetet och vidareutveckling har skett då behov uppstått. Några av anvecklarnas arbetsuppgifter kan sägas vara relaterade till utveckling lika mycket som till användning av systemen (t.ex. Bosse och Olle). Systemen har dessutom oftast varit ganska små och kunnat tillverkas under kort tid. Utvecklingen av systemen har varit relaterade till kalkylprogram som anvecklarna haft viss kunskap om. Dessa faktorer gör tillsammans att TSU inte framstår som något realistiskt alternativ i synnerhet som specialisttjänster kan ses som en begränsad och inte lättillgänglig resurs.

¹ **hybrid** = korsning, bastard (Nationalencyklopedin).

16.2.1 Självförsörjning, gruppanveckling och fadderanveckling

Utveckling av kalkylsystem kan bedrivas av anvecklaren själv eller i grupp. De anvecklade systemen kan användas av anvecklaren/anvecklarna eller av personer som arbetar med samma arbetsuppgifter. Det som gör att det är frågan om anveckling är att den som utvecklar har djup verksamhetskunskap. De olika typerna av anveckling kan kallas självförsörjning, gruppanveckling och fadderanveckling (se Tabell 13 sid 217). I *Del II* och i Avdic (1995a) finns exempel på de olika typerna.

16.2.2 Professionalism och kontroll

Förutsättningar som gör kalkylprogramanveckling lämplig, finns redovisad i kapitel 10.3. Förutom att det skall finnas ett lämpligt verktyg skall anvecklaren själv ha vissa kunskaper och viss attityd. Anvecklarens arbetsituation skall innebära tid till anveckling, kontroll över arbetsituationen, lämpliga arbetsuppgifter, självständiga arbetsuppgifter samt stöd från ledning och dataavdelning. Denna beskrivning kan relateras till Rolfs (1991:114) beskrivning av en profession, som kännetecknas av förekomsten av kompetens. De studerade anvecklarna kan karakteriseras som reflekterande praktiker (Schön 1982:23), med självständiga arbetsuppgifter och med möjlighet att påverka det egna kollektivets regelverk.

Den självständighet som samtliga de studerade anvecklarna åtnjuter kan relateras till de uppfattningar om kontroll och styrning som utmärker olika teoretiska riktningar inom forskningsområdet End User Computing (se kap 10.4.7). I en verksamhet som utmärks av *hög-kontroll-synsättet* kan den professionelle anvecklaren ha svårare att använda KPA för att t.ex. ifrågasätta delar av en verksamhet än i en verksamhet som utmärks av *låg-kontroll-synsättet*. Ledningens och dataavdelningens önskan att styra och kontrollera för detaljerat är inte förenligt med en livaktig KPA-verksamhet. Skillnaden mellan de båda synsätten kan vara att företrädare för *hög-kontroll-synsättet* ser anveckling som en variant av TSU, vilket inte är fallet. Den metodtradition som finns inom TSU är till stora delar inte tillämplig på anveckling. De båda synsätten är till stor del inkommensurabla (se Tabell 12 sid 183).

16.2.3 Integration på flera plan

Kalkylprogramanveckling utmärks av *integration* i flera avseenden (se sid 190). I kalkylbladet integreras systemfunktioner, i anvecklarens arbetsuppgifter integreras dels verksamhetsansvar och systemutvecklingsuppgifter dels olika systemutvecklarroller, kalkylprogrammet integrerar olika programfunktioner etc. Denna integration i flera plan skapar möjlighet till *interaktivitet* i utvecklingsarbetet. Anvecklaren kan snabbt byta mellan utvecklings- och användningsroller samtidigt som ett kontinuerligt lärande sker. Utveckling, användning och lärande blir på detta sätt sammanflätade och integrerade aktiviteter. Anvecklaren utvecklar kunskap om såväl användning av kalkylprogrammet som om problemområdet, vilket kan leda till ifrågasättande av befintlig verksamhet, vilket i sin tur kan leda till fortsatt utveckling (se Figur 110 nedan).



Figur 110 Anveckling innebär integration, interaktivitet och ifrågasättande.

Om anvecklaren utvecklar kunskap om verksamheten kan detta leda till modifiering och/eller nyutveckling av kalkylsystem. Om anvecklaren utvecklar kunskap om verktyg (IT som möjliggörare) kan även detta leda till anveckling. Ett exempel på detta är de intervjuade anvecklarna som använder kalkylprogram för att få aktuellt och noggrant beslutsunderlag. Programmet ger möjligheter till kontroll, formalisering och analys som fungerar som ett stöd för problematisering och ifrågasättande. Slutsatsen av detta är att ökad kunskap om utvecklingsverktyg kan ge ökad kunskap om verksamheten (se Figur 111 nedan).



Figur 111 Konsekvenser av ökad verktygskunskap.

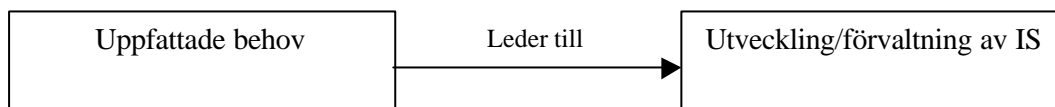
En inte ovanlig synpunkt om anveckling, är att anveckling, i motsats till TSU, inte motverkar ett *tunnelseende*. Tunnelseende kan innebära att en verksamhetsföreträdare inte förmår att se med distans på sin verksamhet och därigenom kan cementera gamla icke-ändamålsenliga lösningar i verksamheten. Denna risk kan knappast avfärdas vare sig när det gäller anveckling, produktionsutveckling, personalutveckling, ekonomi eller någon annan delverksamhet i en organisation. Att få synpunkter från utomstående med andra perspektiv, än den i verksamheten djupt involverade personen, kan ofta vara lämpligt. Anveckling som fenomen förhindrar inte detta. Oavsett om intryck erhålls från utomstående eller ej, är det den verksamhetsansvarige, som avgör vilka åtgärder som kan anses vara de lämpligaste i en verksamhet. Själva invändningen att anvecklare inte vet sitt eget bästa kan ses som ett uttryck för en syn, som i ett annat sammanhang, t.ex. det politiska livet skulle ses som ett uttryck för expertvälde (se även diskussionen om *verksamhetsmotverkande chauvinism*, sid 19).

Integration som den beskrivs ovan kan även ses som stående i motsättning till den princip om arbetsfördelning och koordination som i hög grad utmärker TSU. I själva verket innebär anveckling en motsatt effekt i verksamheter.

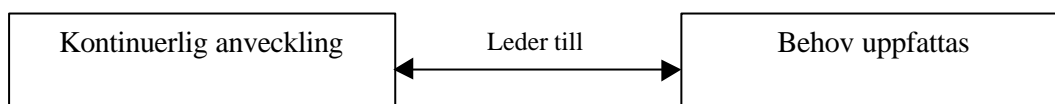
16.2.4 Incitament till anveckling

Initiativet till att anveckla kan antingen uppstå ur behov som uppfattas i verksamheten eller så kan ett uppfattande av kalkylprogrammets möjlighet att importera data initiera idéer som leder till verksamhetsutveckling. I det senare fallet fungerar kalkylprogrammet som möjliggörare. Det första fallet kan ses som ett uttryck för den traditionella ansatsen som representeras av Livscykelmodellen. Implicit i Livscykelmodellen ligger en uppfattning om att behov i verksamheter uppstår, analyseras och tillgodoses, eventuellt med systemutveckling. När nästa behov uppstår, sker analys och eventuell

systemutveckling² igen (se Figur 112 nedan). En delvis motsatt syn representeras av KPA, där anveckling bedrivs kontinuerligt och där utvecklingsarbete inte behöver komma efter upplevda behov, utan kan vara en följd av kontinuerligt ifrågasättande av verksamheten genom anveckling (se Figur 113 nedan).



Figur 112 Traditionell syn behovsuppfattning.



Figur 113 Anvecklingssyn behovsuppfattning.

Den traditionella synen på utveckling är mer statisk än anvecklingssynen på så sätt att traditionellt utvecklade informationssystem inte är avsedda för att förändras kontinuerligt. Ett TSU-ideal är att systemen skall utformas så att förvaltning och nyutveckling minimeras. Enligt denna syn är det förenat med ansträngningar och vissa problem att revidera befintliga system för att bättre motsvara förändringar i verksamheten. I vissa fall kan problem att förändra system inverka på förändringsmöjligheterna.

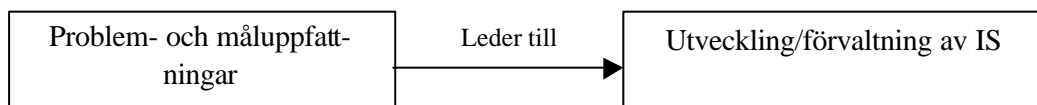
16.2.5 Målstyrd anveckling

Om anvecklarens praktik utmärks av målstyrning snarare än procedurstyrning (se kap 12.3 och 14.3) har anvecklaren stor frihet i sitt val av åtgärder för att uppnå mål. Då handlingar utlöses av strategiska mål, kan dessa mål tolkas olika av olika anvecklare kan innebära att olika åtgärder vidtas beroende på omständigheter. Resultatet från undersökningen *Industri* visar att effekter av anveckling kan vara att anvecklaren kan ta hänsyn till många föränderliga mål och att han kan få information med hög aktualitet och noggrannhet. Detta innebär i sin tur att kalkylsystem kan bidra till bättre beslut därför att diskussionen inför beslutet förs upp på ett högre plan än vad som annars skulle varit fallet. En förutsättning i arbetsmiljön är att anvecklarens arbetsuppgifter är sådana att osäkra beslut skall fattas (se Lämplighetsfaktorer sid 197). En annan faktor är anvecklarens attityd till såväl kalkylprogram som till arbetsuppgifter (se Lämplighetsfaktorer sid 197). Denna faktor låter sig inte enkelt (om alls) mätas, men kan ha stor betydelse för hur lyckad anvecklingsverksamheten blir. De sex intervjuade anvecklarna i kapitel 3 och 4 kan alla beskrivas som engagerade, positiva och självständiga. Ingen av dem var beordrad att skapa kalkylsystem med kalkylprogram, de hade självständigt valt sitt sätt att lösa de arbetsuppgifter som varit orsaken till de utvecklade kalkylsystemen. Detta förhållande stämmer med de egenskaper som en anvecklare förväntas ha (kreativitet och ansvarstagande) för att anveckling skall anses som lämplig (se kap 10.3).

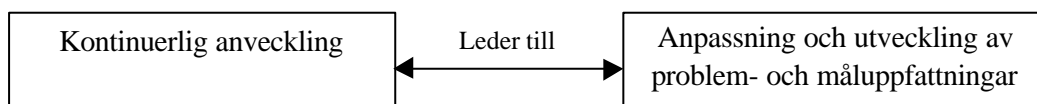
² Eventuellt i form av förvaltning.

En annan förutsättning för nämnda effekter (hänsyn till många föränderliga mål och möjlighet att få information med hög aktualitet och noggrannhet) är kalkylprogrammets integrerade och interaktiva karaktär. Denna integrerade, interaktiva situation innebär att anvecklaren inte är låst i att utföra specifika aktiviteter i vissa lägen utan kan utföra olika aktiviteter utifrån vad som uppfattas relevant för ögonblicket. Anvecklaren har en frihet att anpassa sin problemlösningsprocess och ta hänsyn till idéer som dyker upp. Om problemlösningsprocessen händelsevis skulle vara icke-rationell och icke-linjär kan kalkylprogram i en LAN-miljö stödja denna.

Den traditionella synen på systemutveckling utgår från att problem- och måluppfattningar går att beskriva på ett sätt som innebär att ett meningsfullt informationssystem går att utveckla. Det blivande systemet utformas från den problem- och måluppfattning som var för handen vid konstruktionstillfället (se Figur 114 nedan). I fallet med undersökningen *Industri* implementerades samtidigt med mina undersökningar ett modifierat standardsystem för order, marknad och lager. Systemet löste många problem i verksamheten, bl.a. genom att processorientera informationshanteringen. Ett kalkylsystem skulle definitivt inte kunna ersätta denna typ av system. Däremot erbjuder den anvecklingsaktivitet, som bedrivs av de tre intervjuade anvecklarna en möjlighet till anpassning till förändrade problem- och måluppfattningar, som inte enkelt kan genomföras med det stora systemet. Undersökningsresultatet tyder på att anveckling dessutom bidrar till att utveckla problem- och måluppfattningar (se Figur 115 nedan).



Figur 114 Traditionell syn problem- och måluppfattningar.



Figur 115 Anvecklingssyn problem- och måluppfattningar.

Såväl på industrin som på myndigheten skedde anveckling för att uppfylla befintliga mål i enlighet med befintliga normer. Situationen på myndigheten kan sägas vara något annorlunda i och med att verksamheten styrs av politiskt fattade beslut, vilka inte ändras kontinuerligt.³ På industrin var situationen mer dynamisk, vilket berörs nedan.

16.2.6 Dynamik och förändring

Systemutvecklingens historia har kantats av stor tilltro till möjligheterna att med strukturerade arbetssätt och strukturerade system, lösa problem i verksamheter. Exempel på detta är MIS, SIS-RAS, Cobol, OOA etc. Denna tro på strukturering och meto-

³ *Vår verksamhetsidé* "Vi ska med kompetens, lyhördhet och personligt engagemang, erbjuda örebroarna stadsbyggnadsservice till efterfrågad kvalitet." (Ur Årsredovisning 1997, Stadsbyggnadskontoret Örebro)

ders inneboende kraft har många gånger gett goda resultat och bidragit till positiv utveckling av verksamheter i olika sammanhang. Strukturerade ansatser har haft det gemensamt att man utgått från att verksamheter och information om denna är formaliserbar och att människors arbetssätt kan anpassas till detta. Den ansats som beskrivs här har en delvis annan utgångspunkt. Utgångspunkten är att verkligheten är dynamisk och svår att förutsäga. Utgångspunkten är också att människors arbetssätt kan vara icke-rationellt och icke-strukturerat. Om det i viss utsträckning kan påstås att informationssystem kan vara föråldrade och konstruerade i en tid som inte överensstämmer med den tid de verkar i så gäller detta i än större grad för metoder. Då system utvecklas med metoder tillämpas ofta ett arbetssätt som utformats från erfarenheter som ligger årtal tillbaka i tiden. Detta behöver inte vara en nackdel, men om verkligheten skulle förändras så att förutsättningar, som inte var aktuella vid metodens konstruktion blev aktuella, kan det hända att de utvecklade systemen inte fullt ut kunde anpassas till dessa nya förutsättningar. De anvecklare som studerats inom ramen för detta avhandlingsprojekt (D1 och D2) har inte använt sig av ett strukturerat arbetssätt. Anveckling kan bryta nya vägar bland traditionella nät av systemutveckling och beslutsfattande.

Exemplet Industri kan ses som ett exempel på en dynamisk verksamhet. Det dynamiska ligger i ständigt pågående förändringar i omgivningen. Exempel på sådana förändringar är:

- Förändring av kundsammansättning
- Förändring i produktsammansättning
- Förändring i produktionsteknik
- Förändring i produktionsorganisation
- Förändring i marknadssituation
- Förändring av lagar och bestämmelser

En föränderlig omvärld ställer krav på aktualitet i anvecklarnas analysarbete. Föränderligheten ställer också krav på rimlighetsbedömningar i analysen. Förändringarna skapar nya förutsättningar för verksamheten. Samtliga tre respondenter på industrin underströk vikten av att analysera och söka kunskap för att "ta reda på hur det är", vilket kan tolkas som en strävan efter noggrannhet. Analyserna utmynnar i kunskap om verksamheten som kan bidra till att ompröva de normer som styr anvecklarna. I detta fall kan man tala om strategiska a-system då de kan leda till omprövning av normer i verksamheten (se Figur 41 sid 79).

I nästa kapitel, *Verktygskunskap för anvecklare*, diskuteras vilka konsekvenser innebörden av anveckling kan få för anvecklares arbetssätt.

16.3 Intention

Anvecklingsverksamhet är integrerad med anvecklarens arbetsuppgifter och styrs därför av arbetsuppgifternas syften. Inom ramen för detta kan olika syften urskiljas, vilket diskuteras och analyseras nedan.

16.3.1 Olika syften

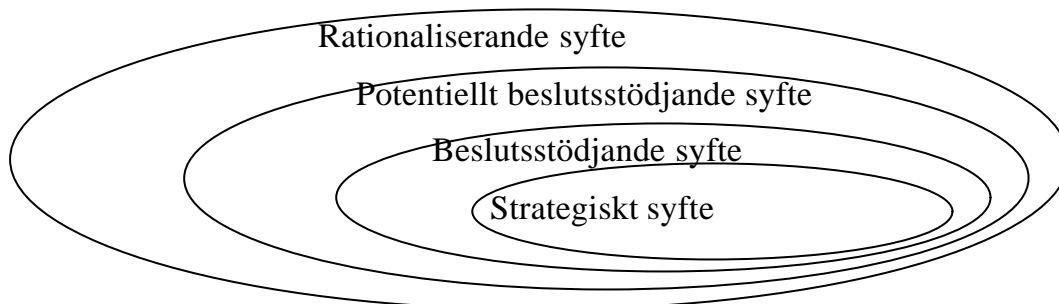
Tre huvudsyften med KPA kan urskiljas: rationaliserande, beslutsstödjande och strategiska (se sid 221ff). De empiriska studierna som gjorts inom ramen för denna avhandling visar att utvecklare använder kalkylprogram av nämnda tre skäl.

Kalkylsystem kan användas för att rationalisera befintliga eller nya (beräknings)uppgifter. Exempel på detta är Olles **dimensionering av bropelare** och Jannes **bokslutssystem**. En automatisering av befintliga rutiner medför mer än enbart rationalisering. Om beräkningarna tidigare utförts manuellt så kommer systemet att tillföras ytterligare funktionalitet i och med att det formaliseras i ett kalkylsystem. Exempel på det är möjlighet till varierad presentation, simulering, distribution, synliggörande av beräkningsgrunder. Exempel på detta visas i Figur 54 och Figur 59. I och med att ny funktionalitet tillkommer så blir kalkylsystemet *potentiellt beslutsstödjande*. Systemet kan bidra till problematisering av frågeställningar av olika slag, vilket i sin tur kan bidra till högre besluts kvalitet.⁴ Några exempel från studierna är mer uppenbart beslutsstödjande, t.ex. Olles **tomtprisberäkningssystem**, eller Thomas **uppföljning av arkningsmaskiner**. I beslutsstödjande system finns normalt även en rationaliserande funktion. Applikationen **distribuerad budget** (se kap 6), rationaliserade en tidigare manuell hantering och bidrog till ökad kvalitet och bättre underlag i beslutsfattande om budget. Kvaliteten ökade i och med att simuleringsfunktionerna i systemet hjälpte såväl kostnadsställansvariga som budgetansvarig att bearbeta information interaktivt i budgeteringsarbetet.

Gränserna mellan de olika syftestyperna är flytande och vilka av systemen i *Del II*, som kan ses som strategiska är inte självklart, men Bosses **täckningsbidrag per timme** skulle kunna sägas ha en strategisk effekt på så vis att det kan påverka synen på hur verksamheten kan betraktas. Ett annat exempel på ett strategiskt system redovisas i Avdic (1995:132ff) där ett företags konkurrenssituation förändras tack vare ett kalkylsystem för offerthantering. Sammantaget kan även en anvecklares anveckling i sin helhet ses som strategisk i det fall som den leder till kontinuerlig omprövning av verksamheten och dess mål. Ett exempel på denna typ av anveckling Ismos anvecklingsverksamhet. Han menar att besluts kvaliteten ökade då kalkylprogramanveckling började praktiseras.

Relationer mellan de olika syftestyperna visas i Figur 116 nedan. Där visas att alla kalkylsystem har ett rationaliserande syfte. Vissa kalkylsystem har dessutom ett potentiellt beslutsstödjande syfte, då endast den rationaliserande effekten ännu uppmärksammas. Några har dessutom ett öppet och aktivt beslutsstödjande syfte. Slutligen har några kalkylsystem utöver, de övriga tre nämnda syftestyperna, ett strategiskt syfte. Det strategiska syftet kan även uppnås genom anveckling som sådan i och med att det kan innebära en generell och långsiktig förändring av informationsbehandlingen i ett företag.

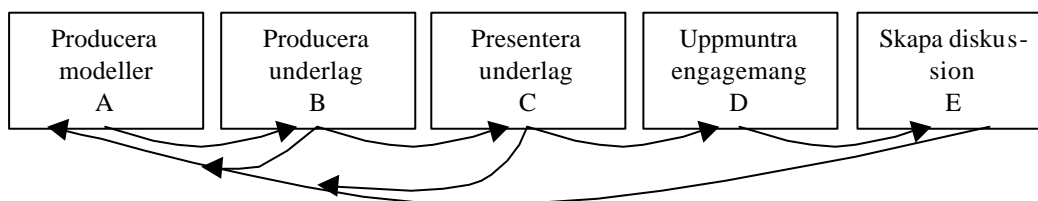
⁴ Det finns även risker och problem med KPA, vilket behandlas nedan.



Figur 116 Olika syftestyper med kalkylsystem/KPA.

16.3.2 Förutsättningar för beslutsstöd

Ett syfte som respondenterna på industrin påtalat har varit att man velat skapa diskussion och uppmuntra engagemang, vilket kan ses som ett delsyfte till beslutsstödjande. I Figur 117 nedan visas flera sammanhängande delsyften som kan relateras till det beslutsstödjande syftet. Dessa delsyften kan ses som relaterade till varann enligt de pilar som finns inritade. Anvecklarens mål kan sägas vara att producera modeller för beslut (A). Modellerna kan användas för att producera underlag för beslut (B). Med systemets hjälp kan underlaget presenteras (C) tack vare den integrerade karaktären på kalkylsystemet.⁵ Underlaget kan i vissa fall användas för att uppmuntra engagemang (D) i syfte att skapa diskussion (E). Det yttersta syftet är att öka besluts kvaliteten. Iterationer förekommer t.ex. genom att de olika delsyftena i sin tur kan leda till förändring eller ny utveckling av modeller. Exempel på kalkylsystem i undersökningen som kan sägas vara främst beslutsstödjande i den bemärkelse som visas i figuren är **uppföljning av arkningsmaskiner och produktionsuppföljning**.



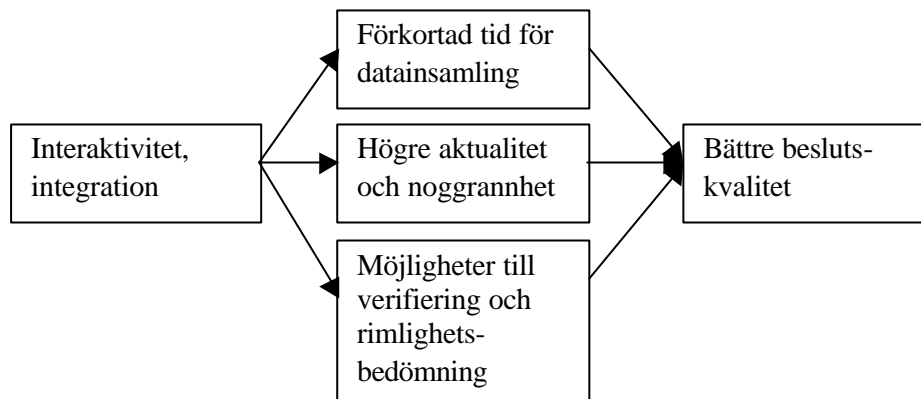
Figur 117 Delsyften till beslutsstöd.

Delsyfte (A) kan ur ett systemutvecklingsperspektiv ses som en form av regelformulering, (B) kan ses som bearbetning, datainsamling och utdatafunktioner, (C), (D) och (E) handlar om överföring och presentation. Speciellt (D) och (E) kan ses som ett uttryck för anvecklarens önskan att skapa en dialog med berörda medarbetare i syfte att erhålla förbättrat beslutsstöd för beslut om verksamheten.

Ett övergripande inslag när det gäller anveckling i beslutsstödjande syfte är strävan efter noggrannhet och aktualitet samt medvetenhet om möjlighet till granskning och verifiering genom rimlighetsbedömning. Då anvecklaren själv utformar det beslutsstödjande kalkylsystemet kan det antas att tiden för insamling av data för beslut blir

⁵ Här avses integration av inmatning, bearbetning, lagring och presentation.

kortare än om beslutsstödssystemet skulle utformas enligt mer traditionella ansatser.⁶ Om KPA innebär att beslut kan tas snabbare eller om mer information kan insamlas till samma beslutstillfälle innebär detta att högre besluts kvalitet kan erhållas i enlighet med Langefors teori om exekutivt effektiv information (se sid 223).



Figur 118 Högre besluts kvalitet med anveckling.

Även Hubers modell (se sid 228) för hur tillgång till IT kan förbättra kvalitativ effektivitet i kunskapsutveckling och beslutsfattande kan sägas ha relevans för anveckling utifrån de studerade fallen.

16.4 Underlag

I kapitel 10 och 12 har vikten av tillgång till underlag i form av information betonats på olika sätt, se t.ex. sid 198 och sid 230. Information, som används i en anvecklares praktik, kan vara av intern eller extern natur. I exemplen i Avdic (1995a:71;132ff) redovisas kalkylsystem med externt underlag i form av information från databaser. Externa underlag kan förmodas bli vanligare i en värld som kännetecknas av ökad globalisering. Ovan (se Figur 111 sid 296) visas att kalkylprogram kan fungera som möjliggörare då ökad kunskap om kalkylprogram leder till ökad kunskap om verksamheten när interaktiv utveckling av kalkylsystem bedrivs. Ett specialfall av detta är då kalkylprogram medger import av data från (1) *lokala*, (2) *organisationsinterna* eller (3) *organisationsexterna databaser* utan att IT-specialister behöver utveckla rutiner för detta. Med lokala databaser menas här databaser som anvecklaren själv har ansvar eller delansvar för. Med organisationsinterna databaser menas databaser som finns inom företaget/myndigheten, men inte inom anvecklaren's ansvarsområde. Med organisationsexterna databaser menas databaser som finns utanför företaget/myndigheten. En viktig skillnad mellan de olika databastyperna är möjligheten att kommunicera med försörjaren, den som har ansvar för informationen i databasen. Detta förhållande kan jämföras med Nurminens beskrivning av kommunikation, som möjlig enbart då informationsavsändaren finns tillgänglig för att underlätta tolkning av information (se sid 168). I det första fallet (1) är försörjaren anvecklaren själv eller någon medarbetare på anvecklaren's avdelning. I det andra fallet (2) finns försörjaren, inom anvecklaren's or-

⁶ Denna jämförelse gäller för de fall där båda alternativen är möjliga. Att anveckling definitionsmässigt är snabbare i tid räknat än TSU beror på anvecklingens arbetsintegrerade och icke-projektorienterade karaktär. För vidare diskussion av skillnader mellan anveckling och TSU, se även Tabell 12 sid 183.

ganisation medan det i det tredje fallet (3) inte säkert kan utredas vem som är försörjare.⁷ Om t.ex. A-listan på Stockholmsbörsen är underlag i en anvecklares praktik, så kan externa försörjare identifieras på olika nivåer. Dels finns den webbmaster som lägger ut data på nätet och som utformar gränssnittet till webbapplikationen, dels finns personer på börsen, som sammanställer dagens omsättning. Detta innebär en uppdelning i form- och innehållsaspekter eller data och information om man så vill. *Data Warehouse* och *Data Mining* är aktuella applikationstyper där externa data kan användas. Bakom de externa data som analyseras i nämnda applikationstyper finns försörjare som inte självklart kan identifieras. Externförsörjarna kan vara en grupp som växer i omfattning och betydelse. En tänkbar konsekvens av detta är att möjligheten att kvalitetsbedömning av data kan vara svårare än om underlaget är organisationsinternt

En omständighet som har gjort det enklare för anvecklare att få access data än det var tidigare är kalkylprogrammets utveckling (se sid 286). Exempel från MS Excel på funktioner som underlättar access till data är *MS Query* för organisationsinterna data och *Kör webbfråga* för organisationsexterna data.

I relation till vad som ovan hävdats om skillnader mellan anveckling och TSU (kap 16.2.4) kan det hävdas att initiativet till att importera data kan antingen uppstå ur ett informationsbehov som uppfattas i verksamheten eller så kan ett uppfattande av kalkylprogrammets möjlighet att importera data initiera idéer som leder till verksamhetsutveckling. Om anvecklarens praktik utmärks av målstyrning snarare än procedurstyrning (se kap 12.3 och 14.3) kan det vara aktuellt för anvecklaren att själv generera underlag på något sätt, t.ex. genom att hämta externa data enligt ovan. Handlingen i anvecklarens praktik kan alltså utösas av strategiska mål och kräva att olika typer av underlag skapas för att kunna utföra handlingen. Då handlingar utlöses av strategiska mål kan dessa mål tolkas olika av olika anvecklare kan innebära att olika underlag måste skapas som en följd av hur mål tolkas. Även här gäller att möjligheter, t.ex. funktioner för att nå externa data via Internet, kan initiera anvecklingsaktiviteter för att uppfylla strategiska mål, vilket kan ses som ett exempel på KPA som möjliggörare.

16.5 Kunskap

Ett kännetecken på en anvecklare är att han har stor verksamhetskunskap och viss verktygskunskap. Detta innebär att kunskapsbegreppet är centralt för analys av anveckling som fenomen.

16.5.1 Verksamhets-, professions- och verktygskunskap

För att bedriva anveckling krävs kunskap om den verksamhet där anveckling bedrivs, om professionen som anvecklaren utövar i sin huvudpraktik och om verktyg för utveckling av a-system/kalkylsystem. De tre kunskapstyperna kan exemplifieras enligt följande:

⁷ De första två försörjarna kan sägas vara förenliga med HIS-modellens syn på kunskap och kommunikation i och med att de normalt är tillgängliga tillsammans med informationen.

1. Kunskap om verksamhetsdomänen, *verksamhetskunskap*. Denna kunskapsstyp omfattar kunskap om den specifika övergripande verksamheten där flera människor arbetar mot samma övergripande mål. När det gäller t.ex. kartongtillverkning vid den specifika industrin som beskrivs i kapitel 3, kan det röra sig om kunskap om produktionen, om vilka fel som kan uppstå, om vilka planer som företagsledningen har, om vilka som är de viktigaste kunderna etc. Denna kunskap kan vara lika för liknande verksamheter och specifik för den aktuella verksamheten. Kunskapen delas av de människor som arbetar i anslutning till organisationen, d.v.s. verksamhetsföreträdare men även av systemutvecklare, som tillägnar sig kunskap för att kunna bygga system som stöder verksamheten. I vissa fall kan IT-avdelningen ha kunskap om affärsverksamheten som inte specialister på olika verksamhetsrelaterade områden har. Produktionsteknik kan vara ett exempel på detta (se kap 5.2.5).
2. Kunskap om yrket/professionen/aktörsrollen, *professionskunskap*. Aktörer med professionskunskap kan vara ingenjörer, ekonomer, inköpare, konstruktörer, produktionsanalytiker etc. Professionskunskap innefattar kunskaper som definierar professionen, dvs kunskaper som gör det möjligt att utföra arbete. För t.ex. en ekonom kan det gälla kunskap om nyckeltal, budgetering, kontoplaner etc. Professionskunskap innefattar kunskap om normer som gäller för professionen (se kap 13.9) och kan även innefatta verktygskunskap. Det kan t.ex. i konstruktörers yrkeskunskap ingå kunskap om CAD-program. Denna kunskap kan ibland formaliseras till viss del. Ett exempel på detta är desktop-program som till viss del formaliserar kunskaper hos grafikerprofessionen.
3. Kunskap om utvecklingsverktyg, *utvecklingsverktygskunskap*, som är ett specialfall av professionskunskap (2). Denna kunskap innehas huvudsakligen av IT-specialister. Kunskapen kan i viss utsträckning kommuniceras och formaliseras till utvecklingsverktyg, t.ex. kalkylprogram. Kunskapen kan även i viss utsträckning läras av anvecklare och syftar då till att utveckla verktyg i den egna praktiken. Kunskap om kalkylprogram kan ses som en professionskunskap för systemutvecklare.

Verksamhetskunskap och professionskunskap kan sammanfalla om det rör sig om t.ex. konsultverksamhet, eller andra fall av att den professionelle sysslar med kärnverksamheten i en organisation. Ett exempel på det sistnämnda skulle kunna vara en lärare som arbetar i en skola. Alla tre kunskapsstyperna kan sammanfalla om någon t.ex. arbetar som konsult inom allmän kalkylsystemutveckling (ej relaterad till specifik operativ verksamhet). I de fall som praktiken innebär utveckling av verktyg och denna utveckling inte tillhör professionens huvudverksamhet, innebär det att den professionelle behöver kunskap ur samtliga tre kunskapsdomäner. Det kan bero på formaliseringsgraden (se Figur 119 nedan) i verksamhets- och professionskunskapen huruvida en aktör är en potentiell verktygstillverkare⁸ eller ej. Detta kan vara ett annat sätt att formulera en typ av kännetecken på anveckling, dvs. att verksamhets- eller professionskunskapen skall vara formaliserbar i någon utsträckning.

⁸ Verktyg i bemärkelsen kalkylsystem.

16.5.2 Fördelning av kunskap

Avgörande för anvecklarens möjlighet att utöva sin profession är tillgång till *kunskap*. Kunskap om och normer för verksamheten, professionen och systemutveckling finns hos anvecklaren eller i utvecklingsverktyget (t.ex. i kalkylprogrammet). I utvecklingsverktyget har verktygsmakaren⁹ formaliserat kunskap och normer. Anvecklaren kan t.ex. få hjälp med att multiplicera matriser (som kan vara en professionskunskap) genom matrisbehandlingsfunktioner och normer om presentation styr utskrifter av t.ex. pivottabeller. Sett ur detta perspektiv finns normer och viss kunskap även inbyggd i hammare, systemutvecklingsmetoder och grävmaskiner. Kunskap och normer finns även hos anvecklaren. Man kan säga att kunskap och normer fördelas mellan anvecklaren och verktyget/verktygsmakarna. När det gäller kalkylprogram kan man å ena sidan säga att verktyget innehåller mer och mer av utvecklingskunskap och -normer. Om man däremot kategoriserar kunnande och normer efter formaliseringsgrad, kan man å andra sidan säga att vissa typer av kunnande och normer visserligen tas över av verktyget, men att detta inte nödvändigtvis leder till att anvecklaren utarmas på kunnande och möjlighet att leva upp till normer. Om man antar att kunnande kan klassas som olika formaliserbart och att verktyg i någon mening till sin natur innehåller ett slags formaliserad kunskap så avbördas anvecklaren från detta och kan ägna sig åt de uppgifter som (än så länge) är svårformaliserbara.¹⁰ Samma sak gäller med normerna. Verktygets begränsningar stipulerar normerna på så vis att ju mindre verktyget klarar av och ju mer begränsad anvecklaren är i användning av det desto mer styrd blir anvecklaren av de regler/normer som verktyget framtvingar. Med detta resonemang kan det hävdas att ju mer generellt ett verktyg är, dvs. ju mer kunskap som är implementerad i verktyget, desto mer av normer lämnas till anvecklaren. Ökad verktygsgeneralitet innebär följaktligen ökad kunskap och mindre tvingande normer i verktyget.

Den kunskap som implementeras i ett verktyg är kunskap som är tillgänglig, kommunicerbar och formaliserbar. I kapitel 1 (se Figur 2 sid 7) visades att det sker en utveckling mot att allt mer svårformaliserbar kunskap kan formaliseras. Om kalkylprogram tas som exempel så kan t.ex. ett kalkylsystem byggas som beräknar vinsten på en order utifrån ett antal parametrar, t.ex. fraktkostnader, valutakurser, produktionskostnader, lagerkostnader, pris till kund etc.¹¹ Om uppgiften skulle utföras manuellt skulle olika typer av kunskap i det aktuella fallet kunna vara

1. att rent matematiskt räkna rätt
2. att räkna snabbt
3. att kunna beräkna med hög precision (t.ex. med tio decimaler)
4. att känna till reglerna för uträkning
5. att hålla ordning på och komma ihåg alla variabler vid beräkningstillfället
6. att känna till variablernas aktuella värden vid beräkningstillfället
7. att kunna utnyttja tidigare beräkningar
8. att bedöma det rimliga i resultatet
9. att presentera resultatet på ett begripligt sätt för olika målgrupper

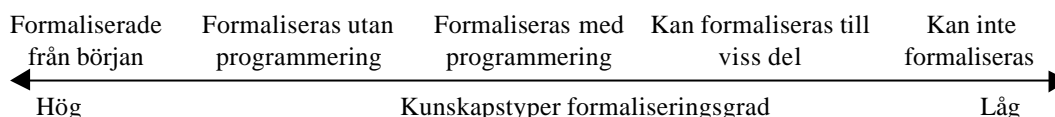
⁹ Som tillverkat t.ex. ett kalkylprogram.

¹⁰ Denna syn på automatisering av kunskap står i viss kontrast till Göranson (1989) som ser automatisering av yrkeskunskap som en risk för utarmning av arbete.

¹¹ Systemet har utvecklats på marknadsavdelningen på *industrin* (se sid Figur 124).

- 10. att kunna modifiera beräkningar utifrån verksamhetens mål
- 11. att kritiskt kunna värdera reglernas relevans.

I exemplet finns vissa kunskapstyper (1,2,3) som i ett kalkylprogram är formaliserade från början. Några kunskapstyper (4,5,6) låter sig formaliseras utan större besvär och utan programmering. Någon kunskapstyp (7) kan formaliseras med hjälp av programmering. Några kunskapstyper (8,9) kan i viss utsträckning formaliseras. Några kunskapstyper (10,11) låter sig inte formaliseras. Här kan man skönja en antydning till kategorisering av kunnande, som kan fördelas på anvecklare och verktyg(s-makare). De olika typerna kan placeras in på en skala av formaliserbarhet.



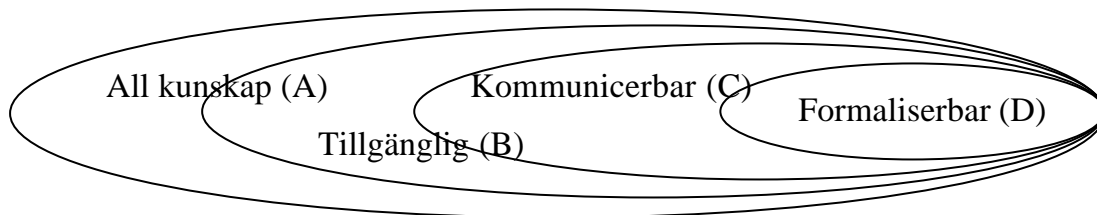
Figur 119 Formaliseringsgrad för olika kunskapstyper.

Ovanstående kunskaper är sådana, som anvecklaren behöver för att utföra sitt arbete. Vissa av dessa kunskaper finns tillgängliga i verktyg, men bara för den som kan använda ett för ändamålet avsett verktyg. Ju större kunskaper om hur verktyg kan användas eller om vilka verktyg som finns, desto mer stöd kan anvecklaren erhålla för utförandet av sina arbetsuppgifter. Detta förhållande till verktyg gäller inte bara anvecklare, det gäller alla som utför arbetsuppgifter med hjälp av verktyg. Ju bättre verktygsanvändaren behärskar verktyget och ju bättre verktyg, desto större stöd kan anvecklaren få för utförande av arbetsuppgifter. Detta förhållande problematiseras vidare i kapitel 16.5.3 nedan.

Inom systemutveckling gäller att TSU genomförs av systemutvecklare i samarbete med verksamhetsföreträdare. Systemutvecklaren har initiativet i utvecklingsprocessen. Systemutvecklaren behöver tillgång till verksamhetskunskap. En del av verksamhetsföreträdarens kunskap är tillgänglig. Anveckling å andra sidan genomförs av anvecklaren med hjälp av utvecklingsverktyg. Anvecklaren behöver viss verktygskunskap. Viss verktygskunskap är tillgänglig för anvecklaren. Denna kunskap kan bli tillgänglig genom utvecklingsverktyg där viss utvecklingskunskap finns formaliserad. Denna situation illustreras i Figur 8 sid 19.

Kunskap om utvecklingsverktyg kan liksom annan professionell kunskap kategoriseras på olika sätt. Den indelningsgrund som kan vara intressant ur ett anvecklingsperspektiv handlar om hur anvecklare kan tillgodogöra sig kunskap om utvecklingsverktyg för att utföra arbetsuppgifter. Den tillgängliga delen av kunskapen om utvecklingsverktyg kan delas upp i formaliserbar och (ännu) icke formaliserbar. Om man tar ett exempel från datastruktureringsområdet så kan teorier om normalisering sägas vara tillgängliga men inte formaliserade i kalkylprogrammet MS Excel. Däremot är möjligheten att ställa SQL-frågor till stor del formaliserad i form av MS Query som följer med MS Excel som ett s.k. tillägsprogram. I Figur 120 nedan visas all kunskap om utvecklingsverktyg som ellips A (som även innefattar B, C och D). Tillgänglig kunskap visas som ellips B (som även innefattar C och D). Kommunikerbar kunskap visas som ellips

C (som även innefattar D) och formaliserbar kunskap som ellips D (se även Figur 1 sid 6).



Figur 120 Indelning av kunskap med avseende på tillgänglighet, kommunicerbarhet och formaliserbarhet.

Utvecklingsverktyget kan också ha kunskaper formaliserade, som hör till andra praktiker men som är relaterade till anvecklarens praktik. När anvecklaren gör ett kalkylsystem med matrisbehandlingsfunktioner så sker inte bara en typ av formalisering. Å ena sidan formaliseras anvecklarens arbetsuppgifter i ett kalkylsystem. Å andra sidan appliceras redan tidigare formaliserade matrisbehandlingsfunktioner, som skulle kunna hänföras till matematikerns kunskapsdomän, på anvecklarens arbetsuppgifter. Den kunskap som finns integrerad i utvecklingsverktyget och sedermera a-systemet i form av funktioner som tillhör andra praktiker, kan utnyttjas av anvecklaren, om han har insikter i hur dessa kunskaper skall utnyttjas. Ett exempel på detta skulle kunna vara regressionsanalys med vars hjälp samband mellan variabler kan analyseras, dock med viss statistikkunskap i botten.

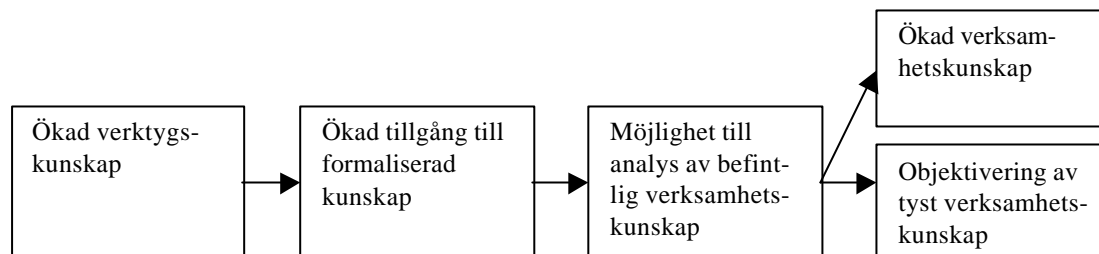
Kalkylprogram har i sig formaliserat kunskap från flera kunskapsdomäner, t.ex. från statistik, företagsekonomi och matematik. Den i verktyget integrerade kunskapen kännetecknas av att den är i någon bemärkelse kvantitativ och därmed formaliserbara

16.5.3 Objektivisering av tyst kunskap

Ovan hävdades att ökad verktygskunskap kan leda till ökad tillgång till den kunskap och de normer som finns formaliserade i ett verktyg. Exempel på detta är de flesta av de system som redovisas i empiridelen, *Del II*. Förutom kunskapen i verktyget kan dessutom ytterligare kunskap erhållas. Ett slags synergieffekt kan erhållas när anvecklarens möjlighet att analysera verksamheten ökar. Denna ökning av analysmöjligheten kan ske på olika sätt. I vissa fall kan det röra sig om ett större system av typen **uppföljning av arkningsmaskiner**, som ger möjlighet till analys och tidigare icke känd kunskap om arkproduktion. I andra fall kan det röra sig om flera mindre system som tillsammans ger möjlighet till analys av verksamheten i olika steg, men som tillsammans kan förbättra kunskapen om verksamheten. Exempel på detta visas i kapitel 3.3. Den förbättrade kunskapen om verksamheten kan dels bestå i resultat av sammanställningar av befintlig kunskap enligt på förhand bestämda förväntningar, t.ex. de system som Annette utvecklat.¹² Dels kan det bestå i vad som skulle kunna kallas objektivisering av tyst kunskap. Denna objektivisering sker när den professionelle anvecklaren

¹² Se avsnitt 4.5. Detta gäller speciellt det ena systemet, **statistik över bygglovsärenden**.

analyserar och bearbetar data på grundval av en idé som inte går ut på ett förväntat resultat, utan där anvecklaren granskar och värderar information för att kunna bilda sig en uppfattning om (delar av) verksamheten. Förhållandet mellan ökad verktygskunskap och ökad verksamhetskunskap visas i Figur 121 nedan.



Figur 121 Förhållandet mellan ökad verktygskunskap och ökad verksamhetskunskap.

Mellan ökad verksamhetskunskap och objektivering av tyst kunskap råder ingen skarp gräns. Objektivering av tyst kunskap är också en form av ökad verksamhetskunskap. Avsikten med att framhäva objektivering av tyst kunskap som en speciell form av fenomen är att fokusera på artikulering av det tidigare icke-artikulerade. Då det gäller rutinmässig sammanställning av befintlig kunskap enligt en fördefinierad mall, t.ex. i en rapportgenerator, skiljer det sig från ett icke-rutinmässigt sökande efter information. Å andra sidan kan rutinmässig sammanställning ge information, som kan ligga till grund för förändrad uppfattning om verksamheten. På samma sätt kan ett fritt sökande efter kunskap leda till att rutiner skapas. I själva verket har alla rutiner en gång skapats genom idéer om hur verksamheten kan dra nytta av information på ett eller annat sätt.

16.5.4 En annan typ av formalisering

Anveckling innebär en annan typ av formalisering än TSU. Ovan argumenteras för att anveckling kan innebära objektivering av tyst kunskap. All tyst kunskap kan dock inte objektiveras. Tyst kunskap kan vara svårformaliserbar och därmed svår eller till och med omöjlig att objektivera. Den finns dock där och kan påverka anvecklarens verksamhet likväl. Hur sker då anveckling med hänsyn till svårformaliserbar kunskap och svårformaliserbara mål? Medan TSU syftar till formalisering av meddelanden mellan aktörer, syftar anveckling till att utnyttja kunskap utan att behöva formalisera den helt och hållet. Hos de studerade anvecklarna utfördes ingen modellering eller systematisering av information om verksamheten i samband med utveckling av olika kalkylsystem. Uppfattade behov översattes direkt till formler, funktioner, texter och grafik på kalkylbladet. Förutom det konkreta kalkylsystemet, t.ex. Ismos **produktionsuppföljning** eller Bosses **täckningsbidrag per timme**, så fanns andra kunskapsaspekter som inverkar på anvecklingen. T.ex. fanns kunskap om verksamheten och dess mål som ledde till att kalkylsystemen gjordes över huvud taget, det fanns kunskap som gjorde att vissa variabler valdes men inte andra, det fanns kunskap som utformade kalkylsystemet layoutmässigt för att informationen skulle presenteras på ett visst sätt. Det fanns även en slags kunskap, som innebar att arbetet med kalkylsystemet ledde till nya insikter om verksamheten. Att analysera de icke-verbala mekanismer som styr vårt agerande i vardagen är inte enkelt. Tyst kunskap är inte påståendekunskap utan ut-

förande- och förståelsekunskap. Rolf refererar Polanyis indelning av aspekter av tyst kunskap (Rolf 1991:64) med vissa reservationer (se kap 13.6.3). Nedan utnyttjas dock dessa aspekter för att försöka beskriva den tysta dimensionen i anvecklarens verksamhet, här i termer av produktionsplanering.

Den *fenomenella aspekten* riktar vår tanke och vår varseblivning. Vad är det i en produktionsplanerarens arbete som styr tanken? Det kan vara en mängd olika insikter, varav en del kan vara medvetna andra omedvetna. I kapitel 3 noterades att produktionsplanerarens arbete hade många mål. Ett intuitivt värderande av de olika målens vikt kan vara ett exempel på hur kringkunskap, uttalad eller ej, påverkar produktionsplanerarens arbete och därmed även de beslut som fattas vad gäller kalkylprogramanveckling.

Ur den *funktionella aspekten* har produktionsplaneraren kunskaper som behövs mer eller mindre direkt för att utföra arbetsuppgifter. Detta är en typ av utförandekunskap som är central i den professionella identiteten för en professionell praktiker. Ur den funktionella aspekten bidrar kunskaper till att veta hur man skall göra och hur man inte skall göra. För en produktionsplanerare innebär denna kunskap en ständigt pågående rimlighetsbedömning.

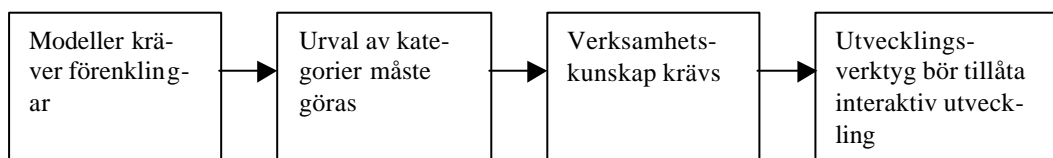
Den *semantiska aspekten* innebär möjligheter till tolkning av innebörder. Sett ur ett anvecklarperspektiv skulle detta kunna innebära att anvecklaren kan bygga kalkylsystem och tolka utdata, utnyttja denna information för att verifiera eller revidera en beräkning eller en sammanställning. Produktionsplaneraren kan utnyttja sitt system för att vinna ny kunskap om sin verksamhet, tack vare sin förmåga att tilldela mening till utdata av olika slag.

Slutligen, den *ontologiska aspekten* av tyst kunskap innebär att tyst kunskap bidrar till att ge en sammanhängande, motsägelsefri och uttömmande beskrivning och förklaring av (någon del av) verkligheten. Den kunskap som behövs för att kunna ha överblick över t.ex. planering av komplex produktion med många motsatta mål är inte möjlig att verbalisera. Det förefaller också otroligt att någon annan än den som på heltid sysslar med denna verksamhet, normalt har en bättre insikt i villkoren för verksamhetens drivande. Då anvecklarens situation jämförs med systemutvecklarens finns det, utifrån ovanstående resonemang, skäl att tro att det normalt finns större möjligheter för anvecklaren att fatta beslut om såväl verksamheten som rörande design av a-system.

Ovan argumenteras för att tyst kunskap utgör en väsentlig del av de resurser som en professionell praktiker, t.ex. en produktionsplanerare, behöver för att utföra sitt arbete professionellt. I och med att kunskapen sägs vara tyst är det inte lätt att granska och kritisera resonemanget. När det gäller de tre anvecklarna på industrin, var de väl medvetna om att den rådande uppfattningen om hur verksamheten fungerade inte var att betrakta som någon "sanning". Det uttalade önskemålet att analysera för att förbättra, kan ses som ett uttryck för att verksamheten inte upplevdes som verbalt kartlagd. Såväl Thomas som Ismo uttrycket en önskan veta "hur det är". Bosse menade att analys var som att "skjuta på rörligt mål". Denna syn visar att anvecklarna inte uppfattar verklig-

heten som entydig och möjlig att beskriva. De ständigt förändrade förutsättningarna gjorde dessutom kraven på aktualitet, noggrannhet och rimlighetsbedömningar väsentliga. Viss kunskap kan också sägas vara tyst i den bemärkelsen att den ännu inte är uppmärksammas. Då exempelvis Thomas strävar efter att hitta orsaker till problem med efterbearbetningen, vet han att det finns en rad ännu okända omständigheter som påverkar verksamheten positivt eller negativt.

I den dynamiska situation, som beskrivs ovan, kan kalkylprogram fungera som en katalysator att hantera den tysta svårformaliserade kunskapen såväl direkt som indirekt. Arbetet med att kartlägga faktorer som påverkade efterbearbetningen (kap 3.5) innebar att historiska data gjordes tillgängliga för analys, vilket i sin tur innebar att Thomas kunde se hur olika förmodade omständigheter påverkade verksamheten. Detta ledde i sin tur fram till nya frågeställningar om hur olika faktorer var relaterade till varann. I dessa fall kan det hävdas att tyst kunskap objektiverades och fördes upp till det fokala planet där det fanns möjlighet att analysera.¹³ Kalkylsystemet skapade möjlighet för diskussion när olika val och deras konsekvenser blir genomskinliga, vilket innebär att ställningstaganden hamnar på en högre nivå än om underlaget inte funnits.¹⁴ I fallet med efterbearbetningen kunde dels valet av faktorer diskuteras och dels kunde de valda faktorernas utveckling jämföras med såväl föregående år som med budget. Valet av faktorer är vare sig självklart eller permanent. En fördel är om verktyget tillåter interaktiv omprövning av modellen (se Figur 122 nedan). Då tyst kunskap objektiveras med kalkylsystem innebär det att verktyget kalkylprogram bidrar till att frigöra tyst kunskap genom att utvecklaren interaktivt kan analysera, bearbeta och påverka verksamheten. I andra fall kan det hävdas att den tysta kunskapen i termer av fenomenella, funktionella, semantisk och ontologiska aspekter kunde spela en roll vid utvecklandet av kalkylsystem på det sätt som beskrivits ovan.



Figur 122 Förutsättningar för formalisering genom interaktiv utveckling.

I en mästare-gesällrelation (se sid 243) förmedlas tyst kunskap från person till person. Vid anveckling kan tyst kunskap användas vid utveckling av kalkylsystem antingen utan att explicitgöras eller genom att explicitgöras och bli kritiserbar. Från att ha varit "Så här har vi alltid gjort" och "Det bara är så" samlas data för analys och bearbetning varvid befintliga uppfattningar, verifieras, revideras eller förkastas. Då utvecklaren analyserar information och finner att den förhärskande uppfattningen stämmer med analysens resultat, kan uppfattningen verifieras och användas som utgångspunkt för verksamhetsuppfattning tills vidare. Då den förhärskande uppfattningen delvis stämmer med analysresultatet, kan beslutsunderlag revideras och användas i sin reviderade

¹³ Denna objektivering har vissa likheter med induktiva expertsystemskal där samband testas för att komma fram till vilka variabler som är mest signifikanta i en viss beslutssituation.

¹⁴ Ytterligare en möjlighet kunde ha varit att utveckla ett traditionellt system med motsvarande funktionalitet. Detta projekt skulle dock blivit betydligt mer omfattande.

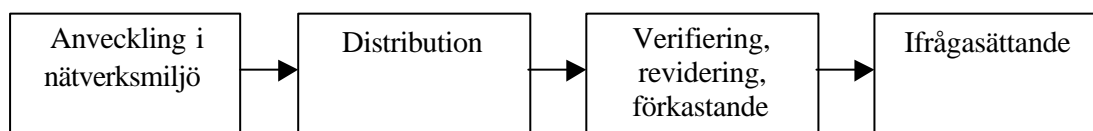
Analys av anvecklingsbegreppet

form. Då den förhärskande uppfattningen inte alls stämmer med anvecklarens analysresultat kan en ny verksamhetsuppfattning etableras och beslut fattas med annat underlag. Denna indelning visas i Tabell 25 nedan.

Tabell 25 Olika konsekvenser av analys av information.

Förhärskande uppfattning	Konsekvens
Stämmer	Verifiering
Stämmer delvis	Revidering
Fel	Förkastande

Möjligheterna att granska information underlättas om fler kan erbjudas möjlighet att granska. Då lokala nätverk (LAN) eller andra nätverk utnyttjas i PC-miljö kan en integration av andra datormiljöer äga rum och det kan bli möjligt för anvecklare att hämta data från avdelnings- eller organisationsinterna databaser eller från externa databaser (se sid 231). Då anveckling bedrivs i nätverksmiljö, tillförs ny funktionalitet form av möjlighet till distribution och verifiering (se Figur 123 nedan). Då anvecklare kan distribuera rapporter som innehåller beräkningar kan såväl resultatet som beräkning som ledde fram till resultatet kontrolleras av berörda medarbetare. Möjligheten till verifiering leder i sin tur till möjlighet till ifrågasättande och möjlighet till förändring av organisationen avseende såväl verksamheten som dess mål (se t.ex. kap 3.4).



Figur 123 Anveckling i nätverksmiljö innebär möjlighet till verifiering och ifrågasättande.

" Den integrerade karaktären på fenomenet anveckling inverkar på hur normer och verktygskunskaper utvecklas och inhämtas. Ett lärande sker genom sammanflätade aktiviteter och den integrerade miljön gör att kunskaper om kalkylprogram kan användas i flera situationer."
(Avdic 1999:326)

- *I detta kapitel formuleras normer för kalkylprogramanveckling*

17 Verktygskunskap och normer

I detta avsnitt diskuteras vilka kunskaper om kalkylprogram som en anvecklare kan vara betjänt av och vilka normer som kan vara relevanta. Utgångspunkterna för diskussionen är behov, möjligheter och problem som kan förknippas med kalkylprogramanveckling. Såväl behov som möjligheter och problem kan härleds ur tidigare redovisade empiriska och teoretiska studier. Som relateringspunkt för analysen används beskrivningen av ett kalkylprograms funktioner (se sid 274) och redogörelsen för anvecklingskunskap i kapitel 13.3.

Exemplet i Figur 124 nedan kommer i detta kapitel att användas för att exemplifiera olika behov, problem och lösningar i detta kapitel. Kalkylsystemet i figuren, som inte omnämns tidigare i avhandlingen, är en modifiering av ett system för offertberäkning. Figuren visar en modifiering av ett kalkylsystem som är utvecklat på det företag som beskrivs i kapitel 3. Motivet till val av exempel är att det innehåller ett flertal aspekter som relateras i kapitlet. Systemet används för att snabbt kunna beräkna en offert med avseende på diverse påverkande faktorer. Systemet är integrerat på det sätt som beskrivs på sid 190. Offertsystemet utgörs av ett antal kalkylblad i en arbetsbok samt några blad i andra arbetsböcker som innehåller bl.a. uppgifter om produktionskostnader, vilka ligger på det lokala nätverket och uppdateras av den person som har ansvar för detta. I figurens nederkant syns de parameterblad (Kund, Artikel, Betalningsvillkor, Valuta, Tillägg och Frakt) som ingår i systemet.

Kalkyl för prissättning		99-03-17	
Data		Kalkyl kr/st	
Kund	SCC	Valutakurs	1,32
Kundnummer	101	Pris (SEK/st)	1584
Fraktklass	4	Frakt %	40
Artikel	Delta	Fraktkostnad	634
Monterad/delar	<input checked="" type="checkbox"/> Kryss=Monterad	Monteringskostnad 1	317
Orderstorlek (antal)	200	Produktionskostnad (mån)	168
Månadsnummer	2	Mängdrabatt	3168
Monteringstillägg (%)	20	Kundorder tillägg	300
Kundorder/lagervara	<input checked="" type="checkbox"/> Kryss=Lagerv.	Resultat 1	3 334
Kundord.kostnad %	15	Administrativ kostnad	158
Betalnings villkor (dagar)	30	Resultat 2	3 175
Offererat pris (lokal valuta/st)	1200	Kundränta	190
Valuta	FRF	Resultat 3	2 985
Kommentarer:			

Figur 124 Ett kalkylsystemexempel.

17.1 Behov och möjligheter

Integration av kunskap från IT-området (se Figur 8 sid 19) innebär en möjlighet för anvecklare att utföra sina arbetsuppgifter bättre än annars. Hur integrationen skall gå till är en central fråga för denna avhandling. Hur kan personer som tillhör en profession ta del av relevanta delar av IT-professionens kunskap och vilka är dessa relevanta delar? Nedan diskuteras hur ett sådant kunskapsbehov skall genereras ur identifierade behov och möjligheter. Utgångspunkt är de resultat som hittills redovisats i *Del II* och *Del III* och som kan tänkas hjälpa till att utpeka kunskapsbehov för anvecklare.

I Avdic (1995a:121) formuleras kunskapsbehov för kalkylsystem med olika komplexitetsgrad som kan ses som en annan dimension vad gäller möjligheter. Dessa kunskapsbehov (se kap 13.3) kan ses som en kunskapsmängd som byggs på ju mer komplext system som skall utvecklas. För de två enklare systemtyperna (*Ordbehandlarsystemet* och *Lilla kalkylsystemet*) krävs kunskaper som de flesta svenskar kan tillägna sig på en förmiddag. För att kunna skapa *Stora kalkylsystemet*, krävs fördjupade verktygskunskaper samt kunskaper om datorns funktion. För det mest komplexa kalkylsystemet (*Applikationen*) krävs förutom nämnda kunskaper, kunskaper i programmering. Sett ur detta perspektiv kan anvecklaren skapa mer komplexa system, oavsett syfte, beroende på hur mycket kunskaper om t.ex. kalkylprogram han tillägnat sig. Olika kunskaper är olika lättillgängliga. Viss kunskap, t.ex. om programmering kan, för att bli användbar kräva mer övning än andra. Annan kunskap, t.ex. om användning av statistiska funktioner, kan kräva bakgrundkunskaper från andra kunskapsområden

som statistik i det nämnda fallet. Pankos indelning i kunskap för anvecklare (se Figur 93 sid 238) har en liknande struktur.

Datastrukturering, specialfunktioner och programmering

Enligt Westerlund (1997) (se kap 13.3) kan anvecklare konstruera mer ändamålsenliga, enklare och mer flexibla kalkylsystem ju högre kunskap om kalkylprogram han har. *Ändamålsenligheten* består i möjlighet att förädla informationen i kalkylsystemet. *Enkelheten* i lägre tid för systemförvaltning. *Flexibiliteten* består i att anvecklaren kan minska antalet konstanter i kalkylsystemet. För att uppnå de eftersträlvade egenskaperna kan anvecklare vidta följande åtgärder: (Westerlund 1997)

1. Logisk uppdelning av likartade data, t.ex. ett kostnadsställe per blad och en avdelning per arbetsbok.
2. Uppdelning av data med olika funktion. I exemplet ovan finns t.ex. indata på vänstersida och utdata på höger sida. Olika typer av underlag finns uppdelade på olika blad.
3. Skydd av delar av kalkylblad som inte skall ändras, t.ex. ledtexter och formler. I exemplet ovan är enbart celler där man skall mata in värden oskyddade.
4. Användning av specialfunktioner, t.ex. finansiella funktioner. I exemplet använd t.ex. funktionen INDEX() för att hitta rätt fraktklass utifrån kundnummer.
5. Användning av funktioner för förenkling av utskrifter, vilka skulle kunna vara anpassning av kolumnbredd, formatering för läsbarhet, procentuell anpassning av utskrift, användning av vyhanterare, användning av rapporthanterare. I exemplet ovan är det område som skall skrivas ut fördefinierat, vilket innebär att just detta område skrivs ut.
6. Användning av tabeller för hantering av större datamängder. I exemplet är allt underlag organiserat i tabeller, vilket underlättar t.ex. sortering, sökning och uppdatering.
7. Redigering av inspelade makron för att undvika långa inspelningar av makro. I exemplet finns inga makron, men i andra fall kan redigeringskunskap spara tid om inspelningen skulle gå fel, eftersom proceduren då inte behöver göras om.

Kunskaper för att utföra ovanstående åtgärder kan kategoriseras som *kunskap om datastrukturering* (1 & 2), *kunskap om ett verktygs specialfunktioner* (3, 4, 5 & 6) och *kunskap om programmering* (7). Dessa tre kunskapsområden skulle följaktligen kunna vara till hjälp för anvecklaren då han vill konstruera ändamålsenliga, enkla och flexibla kalkylsystem.

Presentation, sammanställning och beräkning/simulering

IT-specialisterna i kapitel 5 pekade ut tre användningsområden som väsentliga för anveckling, nämligen presentation, sammanställning (av centrala data) och beräkning/simulering. De empiriska studierna, presenterade i kapitel 3 och 4 visade att kalkylsystemens överskådlighet, tydlighet och flexibilitet ses som viktiga både vad gäller presentation och analys.

Kunskaper som krävs för att kunna presentera data kan delas upp i rena presentationsaspekter och kommunikationsaspekter. För att omvandla data till information, som skall ge avsedd effekt i en verksamhet kan sammanställnings- eller formateringsfunk-

tioner användas, t.ex. pivottabeller eller diagram. I exemplet ovan är kalkylbladet utformat för att framhäva väsentligheter. De formateringsfunktioner som använts är färg, typsnitt, typstorlek, ramar, centrerung och disponering/redigering. Kommunikationsaspekter handlar om hur en tänkt mottagare kan erhålla information. Detta kan åstadkommas genom kunskap om överföring via nätverk eller annat medium t.ex. diskett eller papper (utskriftsfunktioner). Exemplet är främst tänkt för intern användning, varför kommunikationsfunktioner ej varit aktuella.

Kunskaper rörande sammanställning av data kan gälla länkning eller annan samma-ställning av kalkyldata, import/export av datafiler, åtkomst till databaser inom eller utom organisationen. Med länkning eller annan sammanställning menas dels dynamisk relatering av data mellan områden, kalkylblad eller arbetsböcker eller statisk relatering i termer av flyttning eller kopiering av data. Med import/export av datafiler menas möjlighet att använda data lagrat som fil eventuellt i annat format än det som kalkylprogrammet normalt använder. Med åtkomst till databaser menas funktioner för att hämta data från databaser i organisationer t.ex. med MS Excels MS Query. Även åtkomst till data via Internet kan räknas hit. I exemplet har länkning använts för att sammanställa underlag. Länkningen har skett till samma blad, andra blad och till andra arbetsböcker.

Kunskaper för att utföra beräkning/simulering kan innefatta formelkonstruktion, funktionsanvändning och programutveckling men även kunskaper om formatering, hämtning av data, länkning, granskning, diagram och säkerhet. Exemplet ovan är ett typiskt simuleringssystem. För att konstruera systemet och underlätta parametermodifiering har funktionerna INDEX(), LETARAD() och OM() använts. Dessutom har kontrollerna listruta och kryssruta använts för att underlätta simuleringshandlingen.

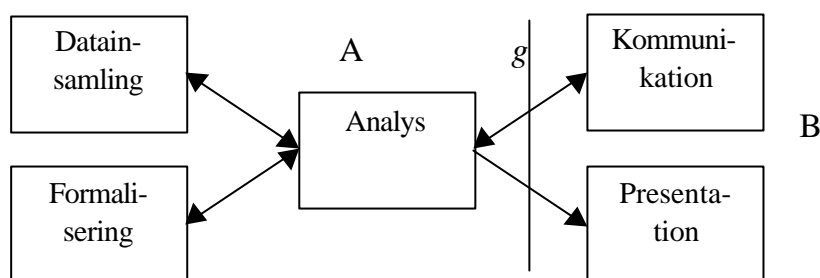
Aktualitet och noggrannhet

De empiriska studierna på industri och myndighet (i kap 3 och 4) pekade på några drivkrafter vid kalkylprogramutveckling som kan relateras till verktygskunskap. Behov som kan relateras till data som används i verksamheter, är aktualitet och noggrannhet. Grundläggande behov hos utvecklaren är möjlighet till datainsamling, formalisering, analys, kommunikation och presentation. Behov som är knutna till noggrannhet och analys är möjligheten till kontroll och återanvändning.

Strävan efter aktualitet och noggrannhet kan ses som mer grundläggande än de övriga eftersom de kan ses som egenskaper hos själva kunskapen som eftersträvas. Med behov av noggrannhet avses här information som är precis, relevant och sanningsenlig i motsats till ungefärlig, traditionsgrundad och osäker. För att samla in noggrann (och annan) information, som inte finns tillgänglig, krävs kommunikation. Denna noggranna information kan sedan analyseras. Analysen kräver först någon form av formalisering. Analysen underlättas av noggrannhet, överskådlighet, tydlighet, flexibilitet och möjlighet till återanvändning. Verifiering av indata och analysresultat sker genom kontroll. Slutligen sker kommunicering av resultat genom presentation och kommunikation.

Huvudaktiviteter vid anveckling

Tack vare anvecklingens integrerade och icke-sekventiella karaktär utförs olika anvecklingsaktiviteter i olika ordning eller samtidigt. Fem huvudaktiviteter skulle dock kunna identifieras utifrån de identifierade behoven. Aktiviteterna är datainsamling, formalisering, analys, kommunikation och presentation (se Figur 125 nedan). Aktiviteterna i området A, till vänster om linjen g, är förbehållet anvecklaren själv medan aktiviteterna i området B, till höger om linjen g, innebär kommunikation med andra intressenter.



Figur 125 Anvecklingsaktiviteter.

Relaterat till indelning i kalkylprogrammets funktioner (se sid 274) kan olika kunskaper relateras till olika behov (se Tabell 26 nedan) och aktiviteter (se Tabell 27 nedan). Indelningarna är översiktliga och visar att den integrerade och interaktiva karaktären på KPA medför att olika kunskaper behövs för att tillgodose olika behov.

Tabell 26 Kalkylprogramfunktioner relaterade till olika behov.

Behov	Relaterade funktioner i kalkylprogram
Aktualitet	Indata, relatering
Noggrannhet	Bearbetning, relatering, analys, operatörsgränssnitt, applikationsutformning, hantering
Återanvändning	Hantering
Kontroll	Bearbetning, relatering, analys, operatörsgränssnitt, applikationsutformning, hantering
Överskådlighet	Relatering, analys
Tydlighet	Relatering, analys, presentation
Flexibilitet	Bearbetning, relatering, analys, hantering, förvaltning

Tabell 27 Kalkylprogramfunktioner relaterade till olika aktiviteter.

Aktiviteter	Relaterade funktioner i kalkylprogram
Datainsamling	Indata, relatering
Formalisering	Bearbetning, hantering, applikationsutformning, förvaltning,
Analys	Bearbetning, relatering, analys, hantering, applikationsutformning
Presentation	Presentation
Kommunikation	Indata, presentation, applikationsutformning

De identifierade aktiviteterna och behoven kan således ses som intimt relaterade till kalkylprogramfunktionerna. Kunskaper om bearbetning, relatering och hantering kan bidra till att tillgodose flertalet aktiviteter och behov. Nedan utvecklas kunskapsbehoven som är relaterade till de olika behoven mer ingående. Aktiviteterna Presentation, Överskådlighet och Tydlighet behandlas i kapitel 17.2.6. Behoven Återanvändning och Flexibilitet behandlas i kapitel 17.2.4. Övriga aktiviteter och behov behandlas nedan.

17.1.1 Aktualitet

Aktualitetsbehovet kan ses som en önskan att arbeta med färskast möjliga data så att analyser grundas på ett underlag som förändras i takt med verkligheten. Detta behov blir starkare ju mer föränderlig omvärlden är. Aktualitetsbehovet kan (1) leda till funktioner för att hålla underlag i ett befintligt system genom länkar eller program som uppdaterar indata. Aktualitetsbehovet kan också ses som (2) en önskan om att kunna hämta aktuell information från databaser e.d. Kunskap om *redigering* är den mest elementära kunskapen med vars hjälp utvecklaren kan få tillgång till data via textfiler, webbdokument eller andra databärare. För att kunna upprätthålla den första formen av aktualitet (1) kan det krävas kunskap om *länkning*, för att länka områden till varandra eller kalkylsystem till kalkylblad som innehåller uppdaterade data och låta dessa vara tids- och datumkänsliga. Om åtkomstrutinerna är komplexa kan viss kunskap om *programmering* behövas. För att kunna upprätthålla den andra formen av aktualitet (2) krävs kunskap om relatering till organisationsinterna och organisationsexterna databaser med hjälp av (SQL-orienterade) *frågeverktyg* eller *webbverktyg*.

17.1.2 Noggrannhet

Det andra grundläggande behovet vid KPA är noggrannhet. Ovan beskrevs noggrannhetsbehovet som en strävan efter precision, relevans och sanningsenlighet. Detta kan gälla såväl själva modellen/kalkylsystemet som informationen som sådan. För att tillgodose noggrannhetsbehovet krävs kunskap om formalisering, analys, kontroll, vilket underlättas av överskådlighet, tydlighet och flexibilitet. Dessa behov behandlas nedan.

17.1.3 Formalisering och analys

Möjlighet till formalisering i form av modeller/kalkylsystem, som kan användas för analys, är en grundläggande egenskap hos kalkylprogram. Denna möjlighet att analysera underlättas av kunskap om – och olika *funktioner* samt om verktyg som ingår i kalkylprogrammet, t.ex. *Problemlösaren* i MS Excel. Analys kan utföras många olika sätt och kan därför underlättas av kunskaper av olika slag (se Tabell 26 ovan). Kontroll (se nästa avsnitt) kan ses som en del av analysarbetet. Överskådlighet (se kap 17.2.6) underlättar möjligheterna att analysera.

Några specifika analyskunskaper som anvecklaren kan vara betjänt av skulle kunna vara simulering, grafisk analys, sammanställningsfunktioner, hanteringsfunktioner och programmering/makro.

Simulering användes t.ex. av Olle för att beräkna tomtpriser (se Figur 51 sid 95). För att skapa en simuleringsmodell krävs kunskaper i formelkonstruktion och kalkylbladsfunktioner. Modellen ska vara flexibel (se kap 17.2.4). *Grafisk analys* kan utföras med hjälp av diagram där olika samband, fördelningar eller trender fokuseras. En speciell form är grafisk analys med punktdiagram där statistiska mått på samband kan genereras. Statistik kunskaper krävs dock för att tolka utdata. Många gånger kan okomplicerade *sammanställningar* som sortering av en tabell vara tillräckligt för att kunna utföra meningsfulla analyser. Mer sofistikerade sammanställningar kan innebära urval, summering, klassificering, andelsberäkning etc. I MS Excel finns t.ex. funktionerna Filter, Delsumma och Pivottabeller, med vars hjälp anvecklaren kan göra analyser och beräkningar av olika slag. Kunskaper om *hantering* och *redigering* av kalkylbladet i stort kan spela en viktig roll för anvecklaren i själva analysituationen. Exempel från MS Excel på sådana hanteringskunskaper är snabbkopiering, gruppredigering, sök och ersätt, serier, datumhantering, klistra in special, olika sätt att markera, matrishantering, operatörer etc. *Programmering* och *makroinspelning* kan dels skapa möjligheter till analyser som annars vore omöjliga och dels kan det bidra till förenklad hantering av kalkylsystem. Ett exempel på det förstnämnda användningsområdet är utvecklandet av en funktion för generering av veckonummer utifrån datum i ett företag som har veckovis uppföljning av produktion. Rutinen programmerades som en funktion och kunde sedan användas som vilken kalkylbladsfunktion som helst. Ett exempel på den andra typen av användning är inspelade makron för att skydda och ta bort skydd på kalkylblad (se Figur 126 nedan). Rutinerna kan sedan om så önskas kopplas till knappar i verktygsfältet för att kunna användas vid behov (se Figur 127 nedan).

```
Sub TaBortSkydd()  
    ActiveSheet.Unprotect  
End Sub  
Sub Skydda()  
    ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True  
End Sub
```

Figur 126 Inspelade makron för att ta bort och sätta dit skydd.



Figur 127 Knappar kopplade till makron i Figur 126.

17.1.4 Kontroll

Kontroll är ett viktigt behov då noggrannhet skall erhållas. Kontroll skulle kunna delas upp i *logisk kontroll* och *rimlighetsbedömning*. Den numeriska kontrollen bygger på kontroll att information är uträknad på ett matematiskt och logiskt korrekt sätt. Denna typ av kontrollfunktion kan stödjas av ett kalkylprogram, vilket visas nedan. Rimlighetsbedömning bygger på verksamhetskunskaper och kan i mindre utsträckning direkt stödjas av kalkylprogram. Däremot kan rimlighetsbedömningar stödjas av logiska kontroller, (se Figur 128 nedan) vilket innebär att KPA kan underlätta rimlighetsbedömning.

Logiska interaktiva kontroller som kan utföras med kalkylprogram skulle kunna vara kalkylprogram som är tillfälliga eller permanenta och som skulle kunna skapas i samma blad, samma arbetsbok eller i annan arbetsbok. I Tabell 28 nedan visas olika kombinationer av kalkylsystem för logisk kontroll. Ju längre ner till höger som kalkylsystemet kan placeras, ju högre interaktivitet kan kalkylsystemet sägas ha.

Tabell 28 Olika typer av kalkylsystem för logiska kontroller.

Kalkylsystem för kontroll	Permanent	Tillfälligt
I annan arbetsbok		
På annat blad		
På samma blad		

I Figur 128 nedan visas ett exempel en tillfällig kontrollberäkning på samma kalkylblad som det kalkylsystem som kontrollberäkningen avser att stödja. Beräkningen är gjord för att rimlighetsbedöma ett värde. Gränsen mellan ett kalkylsystem för kontroll och andra kalkylsystem kan vara flytande. I själva verket skapas kalkylsystem inte sällan för just kontroll, t.ex. **kostnadsjämförelse för arkning** (se Figur 32 sid 70). Även stora kalkylsystem kan vara tillfälliga i den meningen att de används vid endast ett tillfälle. Den typ av beslutsstödssystem som Stabell (1987:243) kallar Decision Analysis är tänkta för att stödja ett enstaka beslut.

	A	B	C	D
1	Råvaror	Förbrukning		
2	Råvara 1	10 000		
3	Råvara 2	900		6,72%
4	Råvara 3	2 500		
5		13 400		

Figur 128 Exempel på kontroll av en råvaras andel (formaterad till procent) av totalsumma.

I kapitel 15.2, klassificeras de funktioner som ett kalkylprogram erbjuder (se Figur 104 sid 275). För att kunna skapa interaktiva kontrollsystem krävs såväl kunskaper om funktioner för hantering som funktioner för interaktiv regelformulering. Kunskap om hanteringsfunktioner är berör redigering, formatering¹ samt användning av guider, genvägar & specialfaciliteter. Kunskaper om funktioner för interaktiv regelformulering berör kunskap om formler, funktioner, matrishantering, makro, värden/formler, referenser/namn, länkning mm.

Kunskaper om formatering kan också bidra till ökad möjlighet av kontroll. Ett exempel på detta är funktionen *Villkorsstyrd formatering* i MS Excel 97, där formatering av celler varnar när värden ej överensstämmer med fördefinierade intervall. En annan typ av kontroll är *granskning* av själva kalkylsystemet. För att utföra detta krävs kunskaper om formelkonstruktion eller granskningsfunktioner.

17.1.5 Kommunikation

Ovan har argumenterats för att ökade möjligheter till kommunikation leder till ökade möjligheter till distribution och därmed ifrågasättande. Av detta skäl bör kunskap om den logiska uppbyggnaden av kataloger och enheter i ett nätverk stödja anvecklarens möjlighet att både distribuera och hämta information. Då det gäller skapande av gemensamma kataloger i ett lokalt nätverk är det en uppgift för en systemadministratör. Än så länge (september 1999) är detta inte en uppgift för anvecklare. För att hämta information för vidare bearbetning krävs olika kunskaper beroende på varifrån informationen hämtas. Aktuella kunskaper kan gälla import av filer av olika typer eller formulering av frågor mot databaser, t.ex. med frågeverktyg och SQL-frågor. En förutsättning för kommunikation är ett lokalt nätverk.

17.2 Det goda kalkylsystemet

I Tabell 22, Tabell 23 och Tabell 24 (sid 278ff) redovisas ett antal risker som är förknippade med KPA och hur dessa kan lösas. I samband med diskussionen kring riskerna, hävdas att domänkunskap, kunskap om design av system samt kunskap om verktyget kan bidra till att förebygga eller åtgärda fel som uppstår. Testning, granskning samt användning av hjälpsystemet är konkreta åtgärder som kan tillgripas. Testning berörs på sid 279 och granskning berörs i kapitel 17.1.4 samt i viss mån nedan.

I *Del II*, framförallt i kapitel 3, 4 och 5 framkom ytterligare några problem med KPA som sammanfattas och redovisas nedan tillsammans med diskussion om vilken kunskap som behövs för att minska dessa problem.

¹ Här avses inte formatering för presentationsändamål, utan formatering för hantering av systemet. Exempel på detta är att kunna ge vissa delar av ett kalkylblad viss färg för att markera dess status.

17.2.1 Ett strukturerat system

För den som blivit alltmer avancerad i sitt anvecklande, blir omfattande länkning mellan celler, blad och dokument ibland rörig. Ostrukturerade system kan leda till svårigheter att använda och förvalta systemet. System kan vara ostrukturerade på grund av att (1) data är ogenomtänkt disponerade på/mellan kalkylblad eller att (2) länkning mellan celler, blad eller arbetsböcker är oöverskådlig och svår att följa. En (3) ogenomtänkt indelning eller avsaknad av kataloger kan leda till att dokument är svåra att hitta och hantera.

Lösningar på struktureringsproblemet (1) är att fördela data logiskt på blad, mellan blad eller mellan arbetsböcker. Fördelning på blad kan t.ex. delas upp så att de delar som är till för inmatning placeras på ett område medan områden som endast visar data samlas på ett annat område. För att relatera data kan länkning användas. Det finns dock en risk att länkning kan leda till ostrukturerade system (se kap 17.2.5). Länkning kan vid genomtänkt användning bidra till att skapa struktur och beräkningskapacitet i kalkylsystemen. Genomtänkt användning innebär att länkning används då länkstrukturen kan bli överskådlig, enhetlig och entydig. I exemplet i Figur 124 finns en ambition att fördela data enligt ovanstående principer. På själva inmatningsbladet är textfält, inmatningsfält och beräkningsfält markerade med olika färger.

För att kartlägga och reda ut ostrukturerade länksystem (2) kan granskningsfunktionen användas. Med hjälp av granskning kan befintliga länkkedjor kartläggas och omstruktureras. Viss granskning är även möjlig med de grafiska effekter som visas då en cell aktiveras och som innebär att de celler som är länkade till den aktiva visas med olika färger.

För att lösa problemet med avsaknad eller ostrukturerad katalogadministration (3), kan upprättande av enhets och katalogsystem göras. Ett val som måste göras här är vilken indelningsgrund som skall användas.

17.2.2 Ett säkert system

Kalkylprogram är öppna system på så sätt att utvecklaren kan skapa en miljö som är påverkbar av användaren av systemet. Jämfört med t.ex. databashanteringsprogram är utvecklingsmiljön mindre styrd, vilket medför viss frihet vid såväl utveckling som användning men även risker som kan innebära att kalkylsystemet kan förändras både avsiktligt och oavsiktligt.

För att förhindra avsiktlig och oavsiktlig förändring av systemet, speciellt då kalkylsystemet används av andra än utvecklaren, kan säkerhetsfunktioner användas. Det vanligaste sättet att öka säkerheten är att skydda de delar av celler, blad och dokument, som inte skall förändras vid användning av systemet. Vid makroprogrammering kan programmeringskunskaper vara till nytta för att kunna redigera inspelade program så att de inte kraschar vid förändringar i omgivningen.

17.2.3 Ett dokumenterat system

Avsaknad av dokumentation av kalkylsystem kan, om systemet är komplext, försvåra användning och förvaltning. Om systemet används mindre frekvent kan användardokumentation behövas även om utvecklaren är användare. Dokumentation kan integreras i systemet genom att den formuleras som kommentarer till celler, som text på kalkylblad t.ex. i anslutning till inmatning eller beräkning eller som egna blad enbart för dokumentation.

17.2.4 Ett flexibelt system

Ett icke ovanligt önskemål på kalkylsystem är att de är flexibla. Ett flexibelt kalkylsystem är ett system som anpassar sig efter nya indata och som är lätt att ändra. Systemet kan vara svårt att ändra om det är ostrukturerat eller komplext (se respektive avsnitt). Systemet kan också vara statiskt om inspelade makron finns integrerade i systemet. Ju mer omfattande dessa makron är desto mer statiskt blir systemet om inte utvecklaren har kunskaper om makroprogrammering. Inspelning av makron kan vara ett sätt att implementera funktioner i kalkylsystemet som annars vore svåra att skapa. Risken är dock att då systemet skall ändras att inspelningen måste göras om.

Användning av referensformler kan bidra till att öka flexibiliteten i ett system. INDEX(), LETARAD() och OM() är exempel på sådana formler. I referensexemplet, i Figur 124 används INDEX() för att hämta data från andra blad. Ett exempel på detta är när användaren väljer en artikel och priset hämtas från artikelbladet. Funktionen OM() används i exempelsystemet bl.a. för att göra beräkning om monteringskostnad skall göras eller ej beroende på om kryssrutan för montering är markerad.

Även kunskaper om programmering kan bidra till att lösa detta problem. Dock bör den kod som utvecklaren skriver dokumenteras för att göra systemet mindre personberoende. Om utvecklaren saknar programmeringskunskaper kan *makroinspelning* användas.

Återanvändning av system kan ses som en flexibilitetsaspekt. Genom att skapa *mallar* kan gjorda kalkylsystem utnyttjas i relaterade sammanhang.

17.2.5 Ett komplexitetshanterande system

Länkning kan skapa flexibilitet och struktur i system men (1) länkning kan också göra system oflexibla, ostrukturerade och alltför komplexa om det sker utan eftertanke. Ett annan orsak till alltför komplexa system är (2) användning av komplicerade och långa formler. Mikas erfarenheter, redovisade i kap 5.4 är exempel på detta.

För att undvika komplexitet av den första orsaken (länkning) kan systemet delas upp på olika blad där länkningen sker på likartat sätt mellan bladen. Ett annat sätt är att, där så är möjligt, skapa länkar med funktionerna LETAUPP() och INDEX(), vilket även gör systemet mer flexibilitet (se ovan). Denna taktik har tillämpats på referensexemplet.

Då komplexiteten orsakas av långa formler kan även då LETAUPP() och INDEX() vara en lösning i vissa fall. Ett annat sätt att minska komplexiteten är att använda *namn* i formler, vilket gör formlerna mer lättbegripliga. I referensexemplet kan en formel se ut som följer: =Offererat_pris_lokal_valuta_st*Valutakurs

Ytterligare ett sätt är att minska komplexitet är att dela upp beräkningar i flera steg. Det kan bli lättare att förstå och granska beräkningarna, om de är uppdelade på flera kolumner. Dölj eventuellt beräkningskolumner om dessa stör läsbarheten. I referensexemplet är kolumnerna D och H dolda och används för mellanberäkningar.

I vissa fall kan villkorsformler av typen OM(), ANTAL.OM(), SUMMA.OM() minska behovet av långa formler och öka möjligheten att skapa flexibla system.

Ett sätt att undvika komplexa formler kan vara att skriva program som t.ex. hämtar data istället för länkning. Användning av program kan dock skapa en annan typ av komplexitet och statik i systemet (se kap 17.2.4).

Om länkning ska användas i lite mer omfattande utsträckning, bör detta ske där länkningen blir enhetlig och där data är relativt enhetliga. Omfattande länkning är olämplig där data är omfattande, oenhetliga och samtidigt dynamiska. Om länkning sker i flera dokumentled finns en risk att data inte uppdateras automatisk, vilket kan leda till beslut på felaktiga underlag.

17.2.6 Ett system med utdata med god läsbarhet

Eftersom kalkylsystem integrerar ett informationssystems olika delar, gäller att god läsbarhet för utdata/presentation kommer att gälla även insamlings-, bearbetnings-, och lagringsdelarna hos kalkylsystemet. God läsbarhet på utdata innebär överskådlighet, tydlighet och enkelhet.

I de fall som kalkylsystem innehåller delar som visar information som skall presenteras för någon mottagare, on-line eller på papper, kommer syftet med systemet att få svårare att bli uppfyllt om inte data presenteras på ett relevant sätt. Om t.ex. felvärden visas omotiverat kan dessa undertryckas med FEL-funktioner, t.ex. kan felmeddelandet #Division/0! i ett kalkylblad undvikas med funktionerna OM(ÄRFEL()). Ismos riktlinjer (se sid 70) bygger på läsbarhet för mottagaren. Den del av kalkylsystemet som skall kommuniceras skall enligt Ismo och Bosse vara på en skärmbild eller ett A4. Den visade bilden skall inte ha för många variabler. Det väsentliga skall vara uppenbart för läsaren. Om bladet är större än en bildskärm kan *skärmdelning* och *rubrikläsning* användas.

Layouten skall inte vara överbelastad. Eventuella beräkningskolumner kan döljas. Ett exempel på detta utöver referensexemplet är **taraberäkning** (se Figur 31 sid 69).

För utvecklare kan möjligheten att skapa tilltalande rapporter vara en väsentlig faktor. I de fall som rapporter är beslutsunderlag och föremål för granskning av andra kan

läsbarheten relateras till anveklarens arbetsuppgifter och till hur dessa bedrivs framgångsrikt eller ej. Ett exempel på detta är Olles **pris- och indexutveckling** Figur 52 sid 96 som skapades som beslutsunderlag inför en större investering. I de fall som utdata vinner på att presenteras grafiskt är följaktligen kunskaper om diagramkonstruktion värdefulla.

Då diagram används finns ett antal diagramtyper att välja på. Valet beror på vilken typ av information som skall presenteras: Tidsutveckling, fördelning, kategorijämförelser, samvariation kan lämpligen representeras med olika typer av diagram (Avdic 1997a:111). Vid val av diagramtyp bör även hänsyn tas till kognitiva aspekter. Wilkinson (1989:42ff) visar t.ex. att 3D-cirkeldiagram ger en felaktig bild av förhållandena mellan "tårtbitarna" i diagrammet. Om ett diagram syftar till att åstadkomma diskussion (se t.ex. kap 3.5) bör diagramtyp väljas som väcker intresse t.ex. ett linjediagram som visar en kritisk variabls utveckling över tiden.

17.2.7 Ett system med tillgång till data

Tidigare i detta kapitel har det argumenterats för vikten av att kunna hämta aktuella och korrekta data från t.ex. databaser. Hur anvecklare når centrala företagsdata är dock inte självklart. I företag kan prestanda riskeras om många får tillgång till företagets databaser. För att undvika detta måste relevanta data sammanställas och verktyg för access till data måste tillhandahållas av databasadministratören. Utformningen av nätverksstrukturen kan vara viktig för möjligheter till dataåtkomst.

17.2.8 Allmänna lösningar på problem

Flera problem som funnits i tidigare versioner av kalkylprogram har lösts i och med att *nya versioner* innehållit mer kraftfulla funktioner av olika slag. Ett exempel är möjligheten att göra klassindelade sammanställningar i MS Excel. I och med version 5.0 blev denna funktion, som tidigare krävde programmering, möjlig. Denna möjlighet innebär att anveklaren kan klassindela data för att söka hitta samvariation i olika datakonstellationer.

En annan allmän lösning på problem av olika slag är *utbildning* på verktyg, såväl på programvaruverktyget (kalkylprogrammet) som på annan utrustning eller på metoder. Målgruppsanpassad utbildning som rekommenderades av Mika, innebär anpassning till målgruppens kunskapsdomän utbildning utifrån behov (se Tabell 6 sid 124). Just utbildning befanns av Brancheau & Brown (1993:463) vara den mest effektiva ledningsaktiviteten för att påverka anveckling positivt (se sid 192). Ytterligare ett utbildningsbehov hos anvecklare som framfördes av Mika, Hans och Sten var mer programmeringskunskaper som skulle minska problemen med kalkylprogramanvändning genom att öka anveklarens möjligheter att skapa kalkylsystem med önskad funktionalitet.

17.3 Sammanfattning – Utvecklingsverktyg och normer

Den integrerade karaktären på fenomenet anveckling inverkar på hur normer och verktygskunskaper utvecklas och inhämtas. Ett lärande sker genom sammanflätade aktiviteter och den integrerade miljön gör att kunskaper om kalkylprogram kan användas i flera situationer. För anvecklaren sker ett synergetiskt lärande genom användning och utförande av arbetsuppgifter, dvs genom anveckling. I kapitlet diskuteras olika utvecklingsverktygskunskaper som anvecklaren kan vara betjänt av vid olika tillfällen.

Av åtgärder som ledningen kan vidta för att stödja anveckling är situationsanpassad utbildning vara en åtgärdstyp som förespråkas av flera forskare och praktiker.

”Anvecklare kan utifrån sin verksamhetskunskap använda kalkylprogram, som ett medel för att öka sin kunskap om verksamheten och därmed också kunna ifrågasätta och förändra såväl verksamheten som de normer som styr denna.”

(Anders Avdic 1999:327)

- *I detta kapitel presenteras avhandlingens slutsatser*

18 Sammanfattande slutsatser

I detta kapitel ges sammanfattande svar på forskningsfrågorna varefter den anvecklar-generiska modellen presenteras.

18.1 Svar på forskningsfrågorna

Den övergripande forskningsfrågan för denna avhandling formulerades i kapitel 1:

Vilka nya möjligheter får användare att utföra arbetsuppgifter då de själva kan bygga informationssystem?

Anvecklare kan utifrån sin verksamhetskunskap använda kalkylprogram, som ett medel för att öka sin kunskap om verksamheten och därmed också kunna ifrågasätta och förändra såväl verksamheten som de normer som styr denna.

De specifika forskningsfrågorna besvaras nedan.

- **Hur kan anvecklaren med hjälp av kalkylprogramanveckling ta hänsyn till svårformaliserbar kunskap och uppfylla svårformaliserbara mål?**

Med svårformaliserbar kunskap avses kunskap som är svår att formalisera och kommunicera. Exempel på svårformaliserbar kunskap i denna bemärkelse är yrkes- och verksamhetsetik samt vissa färdighetskunskaper. Den svårformaliserbara kunskapen kan ofta sägas vara tyst kunskap. Med KPA kan man i viss utsträckning formalisera tyst kunskap ibland utan att informationen görs kommunicerbar. Med KPA kan även tyst kunskap utnyttjas utan att den artikuleras eller formaliseras. Anvecklare kan expli-

citgöra tyst kunskap genom att bygga kalkylsystem och på så vis göra den tysta kunskapen explicit, vilket i sin tur gör den möjlig att värdera och kritisera för en vidare grupp intressenter.

I anvecklarens kunskapsarsenal finns en inte alltid uttalad insikt om såväl yrkets som verksamhetens mål. Detta kommer till uttryck i den verksamhet som anvecklaren bedriver i sin dagliga gärning. Kalkylprogrammet fungerar som ett utvecklingsverktyg i anvecklarens verksamhet. När anvecklaren använder utvecklingsverktyget i sitt arbete för att utföra olika arbetsuppgifter, kan insikten om såväl yrkets och verksamhetens mål kan få genomslag även där. Anvecklingsverksamheten är en del av anvecklarens ordinarie verksamhet. Det som gör detta speciellt är att det tidigare inte har varit möjligt för personer i verksamheter att skapa datoriserade informationssystem utan hjälp av utvecklingsspecialister.

- **Vilka utvecklingsverktygskunskaper behövs för att utföra olika arbetsuppgifter?**

Det flexibla och icke-strukturerade sättet som KPA bedrivs på kräver god verksamhetskunskap och ett flexibelt integrerat verktyg (se Figur 122 sid 310). Flexibilitet i verktyget innebär befintlighet och tillgänglighet till funktioner. Denna flexibilitet ger möjlighet att pröva olika lösningar och snabbt växla mellan olika aktörsroller. Ett elementärt konstaterande är att ju större kunskaper om kalkylprogram som anvecklaren har, desto komplexare kalkylsystem kan han anveckla. Det är dock inte säkert att effekten blir större ju större verktygskunskaper anvecklaren har. Ett exempel på detta är Ismo, som gjort flera intressanta kalkylsystem av typen *Lilla kalkylen*. I övrigt diskuteras verktygskunskaper i kapitel 17.

Normer och kunskaper från kunskapsdomänen systemutveckling finns integrerade i kalkylprogrammet¹ (se Figur 74 sid 151). Den kontinuerliga utvecklingen av kalkylprogram gör att mer och mer av dessa normer och kunskaper finns integrerade för varje ny version av kalkylprogrammet, vilket gör att kunskaper blir tillgängliga via verktyget.

- **Vilka övriga förutsättningar påverkar anvecklarens möjligheter att utveckla kalkylsystem?**

Inom EUC-forskning kan två synsätt identifieras vad gäller hur anveckling skall hantteras av en organisationsledning. Hög-kontroll-synsättet förespråkar styrning av anvecklingsverksamheten för att undvika problem i form av ostrukturerad och icke-ändamålsenlig anvecklingsverksamhet. Låg-kontroll-synsättet fokuserar på att positiva effekter kan erhållas om anvecklingsverksamheten inte styrs i alltför hög utsträckning. I de empiriska studier som ligger till grund för denna avhandling finns endast exempel på låg-kontroll-anveckling. Skälet till detta är att de personer som studerats haft självständiga, målstyrda arbetsuppgifter där styrning inte använts på anveckling i högre

¹ Även kunskap och normer från andra kunskapsdomäner, t.ex. statistik och företagsekonomi, finns integrerad i ett kalkylprogram.

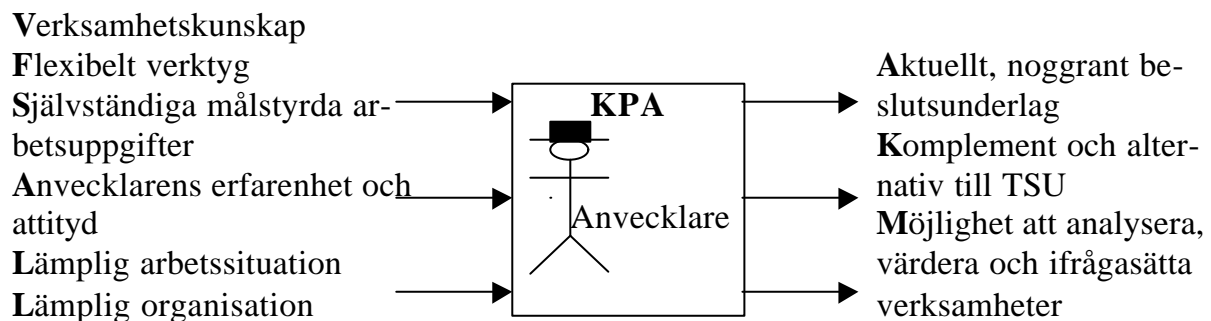
grad än på någon annan arbetsuppgift som anvecklarna haft. Anveckling i den betydelse som den förekommer i denna avhandling kan inte fungera i en organisation med hög-kontroll-synsätt. Förutsättningarna i övrigt har i kapitel 10 klassificerats enligt taxonomin möjligt-berättigat-lämpligt (se kap 10.3). Förutsättningarna kan grupperas enligt anvecklares datorerfarenhet och attityd, lämplig arbetsorganisation samt lämplig omgivning.

- **Vilka effekter får kalkylprogramanveckling (KPA)?**

KPA möjliggör sökande efter noggrannhet, vilket kan ses som ett uttryck för en önskan att öka sin kunskap om den egna verksamheten. Att komma närmare "hur det är". Att gräva djupare. Att kunna analysera vidare. Att få bättre beslutsunderlag. I kapitel 10.4 görs en genomgång av tänkbara effekter av KPA i termer av systemoptimalitet - verksamhetskunskaper, systemkvalitet - systemutvecklingsarbete, informationskvalitet, resursanvändning, beslutsfattande och arbetssituation samt inflytande och styrning.

KPA kan komplettera och ersätta viss TSU. Med KPA kan man utföra viss systemutveckling, som inte går att göra med TSU. Ett exempel på detta är möjligheten att göra permanenta eller tillfälliga kalkylsystem.

En central effekt av KPA är möjligheten att analysera, värdera och ifrågasätta verksamhet. Denna för anveckling centrala effekt redovisas i kapitel 18.3 nedan. Förutsättningar och effekter sammanfattas i Figur 129 nedan.



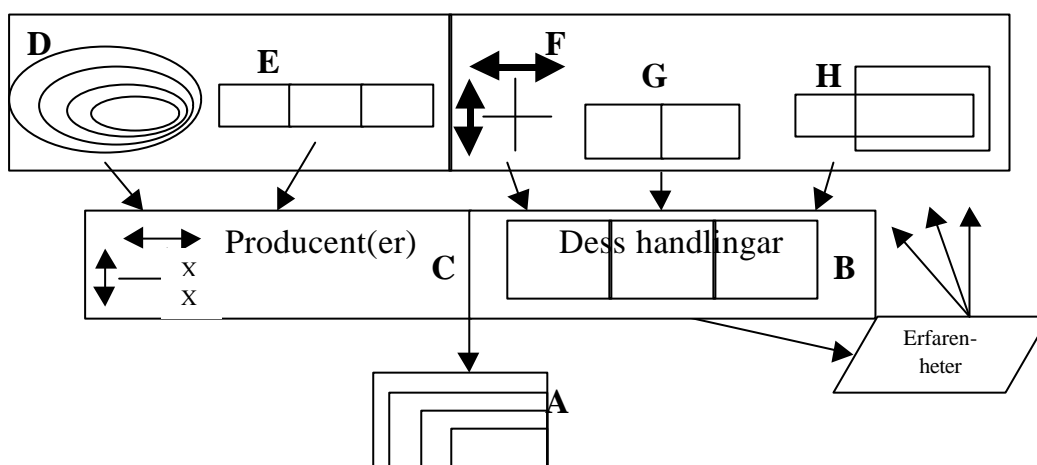
Figur 129 Förutsättningar för och effekter av KPA.

18.2 Den anvecklargeneriska modellen

Ett syfte med avhandlingen är att skapa en begreppsmodell för KPA. Nedan (Figur 130) redovisas en sådan modell med utgångspunkt från den praktikgeneriska modellen. Modellen kallas den anvecklargeneriska modellen. I modellen sammanfattas centrala kategoriseringar som framkommit från delprojekt 1 och 2. Redovisningen följer den praktikgeneriska modellen precedensanalytiskt och referenser görs till tillämpliga delar i avhandlingen. Innebörden i KPA behandlas i kapitel 18.3.

Kalkylsystem (A) kan delas in i Ordbehandlarsystemet, Lilla kalkylsystemet, Stora kalkylsystemet och Applikationen (se Figur 79 sid 164). Anveckling (B) kan ske som

Självförsörjning, Gruppanveckling eller Fadderavveckling (se Tabell 13 sid 217). Anvecklare (C) kännetecknas av stor verksamhetskunskap och viss utvecklingsverktygskunskap (se Figur 7 sid 16). De kallas då standardavvecklare. Om de har stor kunskap om utvecklingsverktyg och stor verksamhetskunskap kan de kallas superavvecklare. Syften med anveckling (D) kan vara rationaliserande, (potentiellt) beslutsstödjande eller strategiskt (se Figur 116 sid 301). Underlag (E) vid anveckling kan vara lokalt, organisationsinternt eller organisationsexternt (se kap 16.4). Kunskap (F) kan gälla verksamheten, professionen eller utvecklingsverktyg (som är ett slags professionskunskap) (se Figur 7 sid 16 och kap 13.1). Normer (G) vid anveckling kan avse verksamheten eller professionen (se kap 1.2.8 och 14.4). Verktuget (H) vid KPA är kalkylprogram i utvecklingspraktiken (se kap 15.2) och kalkylsystem i huvudpraktiken (se kap 8.3). Delar av kalkylprogrammet ingår i kalkylsystemet.



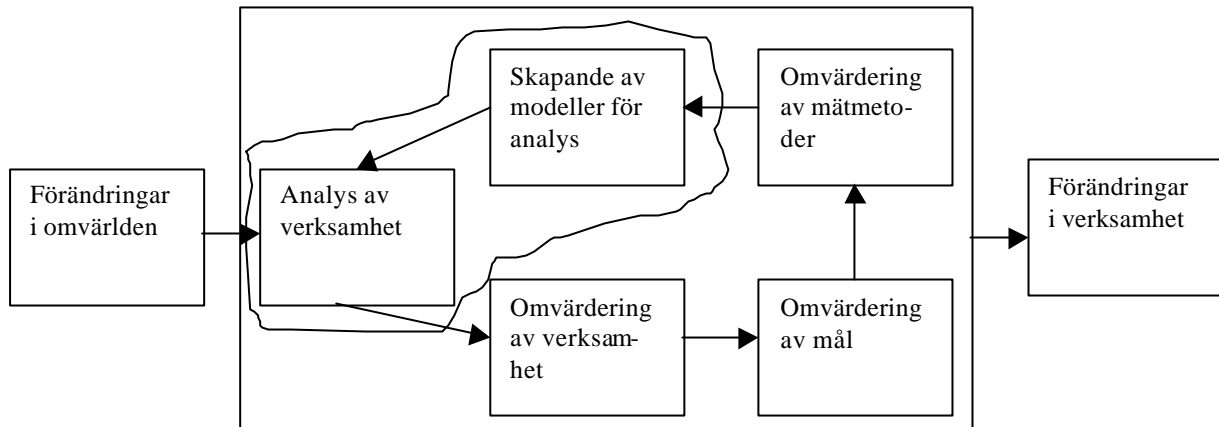
Figur 130 Den anveklargeneriska modellen.

18.3 Kontinuerlig förändring

Ett slutsats i min licentiatavhandling (Avdic 1995a) var att KPA utmärks av arbetsuppgiftsperspektiv, en kalkylprogramorienterad aspekt, en tillgänglighetsaspekt samt decentralisering genom integration (se kap 10.2). Utöver dessa kännetecken har denna avhandling resulterat i ytterligare kännetecken på KPA. Kalkylsystemen i studierna kan överlag ses som beslutsstödssystem men flera av systemen kan ses som uttryck för något mer än bara beslutsstöd. KPA kan ses som ett sätt att möta kontinuerligt förändrade förutsättningar i omgivningen.

Såväl utveckling av kärnverksamheter som utveckling av IT är förändringsfaktorer som inverkar på människors möjligheter att utföra arbetsuppgifter. Förändringssituationen utmärks även av förekomsten av många mål (se sid 70). Verksamheter kan påverkas av förändringar i kundsammansättning, produktsammansättning, produktionsteknik, organisation, marknad och lagar (se kap 16.2.6). IT kan förändras genom utveckling av interaktiva verktyg för analys (t.ex. kalkylprogram), nätverk för distribution av information, och genomskinlig och manipulerbar information. Denna förändring påverkar möjligheterna att analysera verksamheter. Den kontinuerliga föränd-

ringscirkeln i Figur 131 nedan åskådliggör ovan beskrivna förhållanden. De inringade modulerna i figuren visar de aktiviteter som är direkt förknippat med kalkylprogramanvändning



Figur 131 Den kontinuerliga förändringscirkeln.

Sättet att möta förändringar i omvärlden kan sammanfattas i termerna integration, interaktivitet och ifrågasättande. Dessa behandlas var för sig nedan.

18.3.1 Integration

En diskussion om relevans med anveckling, måste främst ses i ljuset av anvecklarens arbetsuppgifter och först i andra hand i ljuset av att anvecklaren bedriver systemutveckling. Systemutvecklingen är integrerad med arbetsuppgifterna. En anvecklares verksamhet kännetecknas av integration av minst två praktiker (se Figur 73 sid 148).

KPA innebär integration av (se sid 190):

- Verksamhetsansvar, verksamhetskunskap och utvecklingskompetens
- Systemutvecklingsaktiviteter
- Aktörsfunktioner
- Behandlingsfunktioner
- Informationssystemets komponenter i kalkylsystemet

Denna flerfaldiga integration skapar möjligheter för anvecklaren att arbeta interaktivt med frågeställningar i den egna verksamheten.

18.3.2 Interaktivitet

Integration medför möjlighet till interaktivitet på flera plan. Interaktiviteten består i att anvecklaren för en dialog med kalkylsystemet, såväl på ett användningsplan som på ett utvecklingsplan. Speciellt interaktiviteten på utvecklingsplanet är en följd av integrationsaspekten. Interaktiviteten i kombination med att anvecklaren i sin huvudpraktik växlar mellan verksamhetsroll och utvecklroll

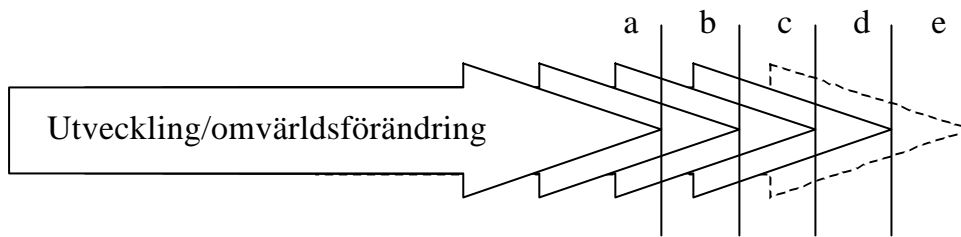
18.3.3 Ifrågasättande

Interaktivitet och integration i utvecklarens verksamhet möjliggör en form av dynamisk och kontinuerlig analys som kan leda till ifrågasättande av den befintliga verksamheten (se Figur 110 sid 296).

Analysen bör utmärkas av relevans, noggrannhet och aktualitet. Till detta krävs möjlighet till rimlighetsbedömning, vilket kräver verksamhetskunskap och viss verktygskunskap. Verksamhetskunskap innefattar verksamhetsetiska hänsyn, vilket är svårformaliserbart, men som ibland kan objektivieras med KPA. Tillgång till aktuella och noggranna data, möjlighet till rimlighetsbedömning och verifiering av information samt kort datainsamlingstid (se Figur 118 sid 302) är viktiga delar i utvecklingsverksamheten. Rimlighetsbedömning och verifiering förutsätter granskning, vilket underlättas av möjligheten att kunna distribuera beslutsunderlag till berörda medarbetare. Förekomsten av ett lokalt nätverk stödjer denna distribution. Ett annat sätt att skapa ifrågasättande är genom diskussion kring grafiskt presenterade modeller. Diskussionen kan sedan bidra till att förändra modellerna av och fördjupa kunskapen om verksamheten.

18.4 Metoder och förändring som permanent tillstånd

Kontinuerlig utveckling innebär både kontinuerlig behovsvärdering, kontinuerlig anpassning och utveckling av problem- och måluppfattningar (se sid 297f). Detta innebär bl.a. att utveckling både kan ses som ett slags potentiellt förstadium till institutionalisering (se sid 247) (Berger, Luckman 1979:34) av (del)verksamheter och som ett uttryck för ett permanent tillstånd av förändring. Kalkylsystem som utvecklare skapar, kan ses som det första stadiet av formalisering av rutiner, vilket sedermera skulle kunna leda till formalisering med TSU. I varje skede kan en organisation ses som en samling formaliserade rutiner och en samling icke-formaliserade (se Figur 132 nedan). Verksamheten i organisationen bedrivs i en dynamisk omgivning som förändras kontinuerligt. För att kunna reagera på förändrade förutsättningar vidtar aktörer i organisationen olika åtgärder. Bland sådana åtgärder kan utveckling av informationssystem finnas. I en situation av ständig förändring kan inte alla informationsproblem lösas med utveckling av TIS. I de fall som TSU tillgrips, kommer en viss eftersläpning att äga rum eftersom systemen tillverkas i en tid vars förutsättningar blir mer eller mindre aktuella allteftersom tiden går. Ett skäl att acceptera utveckling i organisationer är det faktum att omvärldsförändringar tenderar att accelerera i vårt samhälle. Med TSU kan mängder av viktiga informationsproblem hanteras på ett ändamålsenligt sätt, men svårigheten att institutionalisera förändringshantering med metoder och modeller är uppenbar.



Figur 132 Utveckling och anveckling.

Förhållandet mellan TSU och anveckling skiljer sig åt när det gäller att reagera på informationsrelaterade omvärldsförändringar. TSU strävar efter att utveckla genom att använda standardiserade metoder och genom att dela upp och koordinera arbetsuppgifter. Anveckling utmärks av *integration*, *avsaknad av standardiserade metoder*, *sammanflätade aktiviteter* och *implicit förändringsanalys* (Avdic 1995a:180). Anveckling skiljer sig dramatiskt från TSU (se jämförelse KPA och Livscykelmodellen Tabell 12 sid 183). Flexibiliteten i verktyget kalkylprogram leder till integration, flexibilitet och icke-standardisering i utvecklingsarbetet. TSU strävar efter strukturerade och rolluppdelade arbetsformer. Avhandlingen argumenterar för att det finns ett värde i integrerade, informella, icke-standardiserade arbetsformer.

Vanliga invändningar mot anveckling är: risk för tunnelseende, möjligheten att använda IT-specialisten som katalysator försvinner, ineffektivitet och suboptimering i systemutveckling, anvecklade system kan bli osäkra, odokumenterade och ostrukturerade. Förvisso kan alla dessa farhågor besannas. I diskussionen kring verksamhetsmotverkande chauvinism (se sid 19) förs en argumentation kring det principiella i dessa problem. För att ändå minska riskerna för detta redovisas nedan några tänkbara forskningsprojekt inom området anveckling.

18.5 Framtida forskning

De problem som nämns i sista stycket i föregående avsnitt kan delas upp i två grupper. Den ena frågegruppen handlar om att det faktum att det är just anvecklaren som utvecklar är ett problem. Den andra gruppen handlar om att anvecklaren är oprofessionell i IT-specialistprofessionen. I det första fallet kan bara noteras att det inte finns något som hindrar anvecklaren att anlita utomstående personer för att stödja honom i anvecklingsarbetet. Till slut är det ändå anvecklaren, som är den ende som kan avgöra vad som är rätt och fel i en verksamhet. Det är han som ansvarar för verksamheten. Om anvecklaren uppfattar det som värdefullt att rådfråga IT-avdelningen eller externa konsulter så står inte det i strid med anveckling som fenomen. Kring denna frågegrupp skulle empiriska studier kunna göras som utröner huruvida anvecklare är mer eller mindre benägna att rådfråga expertis utanför den egna avdelningen och i så fall varför.

När det gäller den andra frågegruppen rörande anvecklarens agerande i IT-specialistprofessionen finns om inte oräknliga så i alla fall mängder med tänkbara forskningsuppgifter. Tänkbara strategier skulle kunna vara att antingen öka anvecklarens kunskap eller att öka utvecklingsverktygens kapacitet och karaktär. I det första

fallet skulle möjligheten för utvecklare att tillgodogöra sig programmeringskunskaper eller andra formaliseringskunskaper kunna studeras. I det andra fallet kan verktygens utveckling mot att stödja alltmer sofistikerade behov kunna studeras. Exempel på sådan forskning skulle kunna gälla möjligheten till dataåtkomst för beslutsfattare med t.ex. Data Warehouse, Data Mart eller Data Mining. Möjlighet till analys med hjälp av statistik skulle kunna studeras. Hjälpssystemens kapacitet och roll för utvecklaren är också ett potentiellt studieområde. Ytterligare ett utvecklingsområde i denna anda, som ligger i sin linda, men som kan antas öka i betydelse är *webbanveckling*. Vilken utveckling kan tänkas ske vad gäller verktyg och användningsområden för webbanveckling är ett intressant studieområde i en tid då Internetanvändningen formligen exploderar.

19 Summary in English

In the thesis conditions and effects of User Systems Development (USD) using a Spreadsheet Program (SP) are discussed. USD is performed by a User-Developer (UD), a person who acts both a user and a systems developer. Typical for USD is that the UD has profound business understanding and the working tasks related to the Information System (IS) in question, which is called User Developed Application (UDA).

19.1 Introduction

In the early days of computers, expertise was needed in order to use computers. As IT-tools have become more powerful and user friendly, more and more people have been able to use computers and programs as tools when carrying out working tasks. Nowadays it is even possible for persons without special IT-training to construct IS:s that only IT-specialists could have done some years ago. In these theses UD:s use of SP as a systems development tool is elaborated.

The basic research questions in the thesis are the following:

- Which new possibilities can computer users achieve in order to perform working tasks, when they can develop IS:s without help from IT-specialists?

The question can be divided into the following subquestions:

- How can UD:s take not-easily-formalized knowledge into consideration when performing USD?
- What kind of tool knowledge does a UD need in order to perform working tasks?

- What other conditions affect the UD:s possibilities to perform USD ?
- Which effects are the results of USD?

The aim of the theses, besides the aim to answer the questions above, is to formulate a conceptual model where the conditions and effects of USD are categorized.

19.2 Methods

These thesis are the second part (D2) of a larger project, divided in two parts where the first part (D1) resulted in my licentiate theses (Avdic 1995a). The strategy can be described as qualitative, hermeneutic, abductive, and Grounded Theory (GT)-influenced.

The *qualitative* aspect is manifested through the aim of understanding and interpreting the reality of the UD:s. Typical in qualitative methods is triangulation of methods. Methods used are Deep interviews, studies of SP-UDA:s, participant observation (only D2), theoretical studies, survey studies (only D1). The *hermeneutic* approach is manifested through the explicit use of the researchers preknowledge of USD and of the respondents. Interviews and contacts have been frequent. The researcher knowledge of USD has resulted in long discussions of the best way to solve specific problems. This has resulted in an atmosphere where the researcher has gained access to the respondents' situation as UD in a way that would have been impossible if e.g. survey studies had been performed. Change of perspective is also typical in hermeneutic studies. In the empirical studies focus has changed between product and process (SP-UDA and USD). Another change of focus also closely related to USD is the change of focus between working tasks and UDA-development. While the UD is focused on working tasks, the observer has been focused on the development aspect. The *abductive* label is relevant since the studies can be described as a combination of deductive and inductive methods. The research methods can be labeled as *Grounded Theory (GT)* influenced since the empirical research has proceeded the theoretical studies both in the theses and in the mind. Another GT-aspect is the aim of an unbiased data collection situation. The choice of both research questions and respondents has been a result of a specific intention but the data collection has been performed with the awareness of the importance of not knowing what really happens when UD:s perform USD. The data analysis has therefore been performed according to GT-methods (data collection, open coding and selective coding). The coding activities have aimed at finding a key variable. The variable found is "integration". The importance of this is elaborated below.

19.3 Empirical studies

Four empirical studies have been conducted. Two studies have focused on UD:s, interviews are made with three persons in an industry and three persons in a public authority. In the third study three IT-specialists are interviewed and in the fourth study the process of developing a multi-user UDA is described and analyzed.

An Industry

The industry is a multi-goal, dynamic business acting in a keen competitive market. The findings show that the UD:s have an aim to perform tasks through USD by making analysis based on precise and updated information. UD:s need flexible analysis tools together with profound business knowledge in order to evaluate the appropriateness of information produced. The ongoing revaluation of business and its conditions results in revaluation of models and measuring methods. SP can support decisions by making formalized models transparent and criticizable. The formalization results in turn in discussions at a higher level than if values themselves were discussed. Distribution of models in local networks affects analyzing capabilities positively.

A public authority

Activities in the authority are characterized by their public nature. This means that business should be nor profitable nor involve a loss. Other important goals are that the best of the inhabitants always should be taken into consideration and that activities should be carried out with openness. This demand for openness means that basis of decisions should be available and comprehensible. Findings show that USD can help UD:s achieve the goals mentioned.

IT-specialists

The three IT-specialists were one IT-manager, one systems developer and one consultant systems developer. Findings show that problems related with USD were lack of documentation, unstructured applications and limited data processing capabilities in SP. The solutions proposed to these problems were SP-education, SP-version upgrading and more structured UDA:s.

Application

The construction of a budgeting SP-UDA has been studied. The findings show that the construction of a complex, multi-user systems can be initiated, monitored, partly executed, verified, partly administrated, and evaluated by UD:s.

19.4 Framework

Since the theses claim to be GT-influenced, studies of related theory have been highly influenced of the empirical studies. As a framework model (?), the generic practitioners model (Figur 76) is used (Goldkuhl & Röstlinger 1998). The model consists of a set of conditional categories, *assignment, basis of decision, knowledge, norms, and tools*. Other categories are *producers* and their *activities* together with *results* and *experiences*. When a UD develops UDA:s he acts in at least two practices (*praktiker*), the developers practice and the main practice. Each practice has its own set of conditions e.g. knowledge, norms and tools.

Information systems (result)

A UDA is an IS (Figur 78). The difference between a traditional information system (TIS) and a UDA is mainly a question of how it is built. UDA:s are built by UD:s with profound business knowledge, while TIS:s are built by IT-specialists.

Systems development (activities)

Traditional Systems Development (TSD) can be characterized by the notion of the “Life Cycle”. User Systems Development (USD) and TSD are profoundly different in many ways.

Category	TSD	USD
Distribution of tasks	Specialization	Integration
Change analysis	Separation	Integration
Type of system	Complex	Simple
Process	Sequential	Integrated
Durance of the process	Project	Work integrated
Developers	IT-specialists	UD:s
System delivery	When constructed	Work integrated
Environment	Multi-user	Single-user
Tools	Programming skill needed	Direct manipulated or declarative tools
Methods	Designed for large complex multi-user systems	No standardized methods

User Systems Development

Researchers have different perspectives on management and control of USD. Whether USD is adapted in an organization depends on the presence of informal channels of communication and how common training on USD tools is. Some basic conditions (suitable tasks, equipment, knowledge, and certain independence) should be at hand to make USD *possible*. If business and information needs are dynamic, USD can be *justified*. USD is *appropriate* when UD:s also has access to well organized data and get support from management and the IT-department.

When discussing effects of USD, advocates of *high control* recommend (strict) organization of USD-activities. Advocates of *low control* consider USD as time saving and appropriate because of the lack of detailed monitoring.

When discussing working methods, “Traditionals” looks upon TIS as an example of USD. “Organizers” aim to organize USD, preferably with the help of an Information Center. Yet another way to view USD is to regard the specific characteristics of USD as basis. This is done by “Reformers”.

Actors

A UD is a person with profound business knowledge who develops IS:s with tasks and activities the UD is responsible for as basis. The UD integrates several roles when performing USD, e.g. different IT-specialists.

Intention and decision support

The assignment initiating USD often aims to produce basis for decisions. SP-UDA:s are therefore often to be looked upon as decision support systems (DSS). Research on DSS is therefor relevant when discussing USD.

Knowledge

At least two kinds of knowledge are relevant to the UD namely knowledge of tools and knowledge of business. Since the latter is the most important to the UD, the former can be used to deepen the latter. UD:s can make tacit knowledge explicit when developing USD, which in turn makes it possible for others to evaluate and criticize the UDA.

Norms

Norms and knowledge are closely related and sometimes hard to keep apart. The same can be said about norms and goals, concepts that are somewhat overlapping. Norms that are central in the thesis are professional ethics. Professional ethics are crucial to the UD since the professionals' activities are monitored not by procedures but by professional and business ethics. Findings show that investigations made by the UD when performing USD can change organizational norms.

Tools

USD tools are closely related to norms and knowledge, since norms and knowledge are implemented in tools. The main tool when performing SP-USD is of course the SP. The SP is interactive and integrated. It contains a) functions for adapting the development environment, b) functions for editing, formatting etc, and c) functions for interactive development and use. The open nature of the SP can cause different kinds of errors. Business-, professional- and tool knowledge can prevent some of these errors.

19.5 Conclusions

User Systems Development

SP-USD is characterized by *integration* in several dimensions: 1) aspects of IS:s (collecting, storing, processing, and distribution), 2) roles (developer, user, manager), 3) roles of actors in systems development (analyst, programmer, database designer, etc.), and 4) processing functions of the IS. The integrated nature of USD results in *interactivity*. Interactivity means that the UD can change quickly between developing and using the SP-UDA. During the USD-process the UD knowledge about business and USD increases. Since the UD business knowledge increases when performing USD, the UD can analyze and also *question* aspects of business (e.g. production measuring methods).

Knowledge

Access to professional-, business-, and tool knowledge is crucial when developing business supporting IS:s. When performing USD, knowledge is divided between the UD and the tool (SP). Certain kinds of (not too complex) knowledge are formalized into the SP and can be used in the SP-UDA. Other kinds can be formalized by the UD into the SP-UDA. Some kinds of knowledge, (e.g. of critical evaluation of the relevance of formulas) can't be formalized at all. Still this kind of not-easily-formalized (sometimes tacit) knowledge can be taken into consideration when using the UDA, since the UD (with business knowledge) is the user of the system. The findings also

show that goals, not easily formalized, can be taken into consideration when performing USD. Also, USD can make formalization of tacit knowledge possible. This can happen when repeated analysis make the UD aware of circumstances not known before, which in turn make other analysis possible.

The use of local area network makes distribution of models within the organization easier. Since the nature of the SP can make algorithms visible to the reviewer, the discussions regarding the relevance of aspects of the systems more detailed and thereby more reliable.

Because of the integrated nature of USD, learning, using and systems development takes place at the same time. Learning applies to both business and tool. One conclusion of this is that tool training can increase quality of USD. One way of the management to support USD is to initiate and encourage UD-tailored tool training.

In the thesis SP-USD is discussed as a mean to control continuous changes in the environment of the organization. Business analysis (1) can result in revaluation of business (2), which can result in revaluation of goals (3) (and norms), which can result in revaluation of measuring methods (4), which can result in new analytical models (5) (UDA), which can result in new business analysis (1) and so on. USD is discussed as one way to meet change as a permanent business condition, which differs from traditional methods for systems development.

Referenser

Action Technologies, (1993) *Action Workflow Analysis Users Guide*, Action Technologies.

Agnér Sigbo G, (red) (1993) *Fortsättning följer... om kontinuerlig systemutveckling och om människor och datorstöd i jobben*, Carlssons bokförlag, Stockholm.

Angehrn A A, (1991) "Supporting Multi-Criteria Decision Making - New Perspectives and New Systems", *Executive Information Systems*.

Allwood CM, (1990) *Människa-datorinteraktion Ett psykologiskt perspektiv*, Studentlitteratur, Lund.

Alvesson M, Sköldbäck K, (1994) *Tolkning och reflektion - Vetenskapsfilosofi och kvalitativ metod*, Studentlitteratur, Lund.

Andersen E S, (1994) *Systemutveckling - principer, metoder och tekniker*, Studentlitteratur, Lund.

Argyris C, Schön D A, (1978) *Organizational Learning: A Theory of Action Perspective*, Addison Wesley, Reading, Mass.

Argyris C, Schön D A, (1996) *Organizational Learning II - Theory, Method, and Practice*, Addison Wesley, Reading, Mass.

Avdic A, (1995a) *Arbetsintegrerad systemutveckling med kalkylprogram*, Licentiatavhandling, Institutionen för datavetenskap, Linköpings universitet.

Avdic A, (1995b) *Redovisning av studier rörande arbetsintegrerad systemutveckling med kalkylprogram. Studie 0-3*, Högskolan i Örebro.

Avdic A, (1996) *Att använda EXCEL 5*.

Avdic A, (1997a) *Att använda EXCEL 97 del 1*.

Avdic A, (1997b) *Att använda EXCEL 97 del 2*.

Avdic A, (1997c) *Att använda WORD 97 för PM- och uppsatsskrivning*.

Axelsson L, Ortman L, (1985) *Utvecklingshandboken SVEA Modell och metoder för administrativt utvecklingsarbete*, Kommundata, Lund.

Bansler J, (1990) *Systemutveckling Teori och historia i ett skandinaviskt perspektiv*, Studentlitteratur, Lund.

Barker R M, (1995) "The Interaction Between End User Computing Levels And Job Motivation and Job Satisfaction: An Exploratory Study", *Journal of End User Computing*, Vol 7, Nr 3, Summer 1995, sid 12-21.

Barrett M, Walsham G, (1995) "Using IT to Support Business Innovation: A Case Study of the London Insurance Market", *Scandinavian Journal of Information Systems*, Vol 7, Nr2, sid 3-21.

Beekman G, (1997) *Computer Confluence - Exploring Tomorrow's Technology*, Addison Wesley Longman.

Benham H, Delaney M, Luzi A, (1993) "Structured Techniques for Successful End User Spreadsheets", *Journal of End User Computing*, Vol 5, No 2, Spring 1993, page 18-25.

Bento R F, (1995) "Cross-Cultural Teamwork in End User Computing: A Theoretical Model", *Journal of End User Computing*, Vol 7, No 4, Fall 1995, page 4-11.

Berg M, Hultman P-O, (1985) *Systemhandboken Klokbok för systemutvecklare*, Liber.

Berger P L, Luckmann T, (1979) *Kunskapssociologi. Hur individen uppfattar och formar sin sociala verklighet*, Wahlström & Widstrand, Stockholm.

Referenser

Bergner T, Hernaes K, Johansson E, Lundeberg M, Olstedt D, (1983), *Självbetjäнад databehandling - samlevnadsformer*, Institut V.

Bogdan R, Taylor S J, (1972) *Introduction to qualitative methods*, John Wiley, New York.

Bowman B J, Grupe F H, Lund D, Moore W D, (1993) "An Examination of Sources of Support Preferred By End-User Computing Personnel", *Journal of End User Computing*, Vol 5, No 4, Fall 1993, page 4-11.

Bowman B J, Grupe F H, Simkin M G, (1995) "Teaching End-User Applications with Computer-Based Training: Theory and an Empirical Investigation", *Journal of End User Computing*, Vol 7, No 2, Spring 1995, page 12-18.

Brancheau J C, Amoroso D L, (1990) "An Empirical Test for the Expansion-Control Model for Managing End-User Computing", In DeGross Alavi Oppelland (red) *Proceedings of Eleventh International Conference on Information Systems*, December 16-19 1990, Copenhagen, Denmark sid 291-303.

Brancheau J C, Brown C V, (1993) "The Management of End User Computing: Status and Directions", *ACM Computing Surveys*, Vol 25, Nr 4, Dec 1993, sid 437-482.

Capron H L, (1996) *Computers - Tools for an Information Age*, The Benjamin/Cummings Publishing Company Inc.

Carlsson S A, (1993) *A Longitudinal Study of User Developed Decision Support Systems*, Doktorsavhandling, Institutionen för Informatik, Lunds Universitet.

Collste G, (1996) *Inledning till etiken*, Studentlitteratur, Lund.

Conger S A, (1992) "An Innovation Theory of End-User Computing", Ingår i Cotterman W W Senn J A (red) *Challenges and Strategies for Research in Systems Development*, John Wiley.

Courbon J-C, (1995) "User-Centered DSS Design and Implementation", *Implementing Systems for Supporting Management Decisions*.

Dahlbom B, Mathiassen L, (1991) *Struggling with Quality - The Philosophy of Developing Computer Systems*, Report 4, Gothenburg Studies in Information Systems, Department of Computer Sciences, Chalmers University of Technology and the University of Göteborg.

Davenport T H, (1993) *Process Innovation - Reengineering Work through Information Technology*, Harvard Business School Press, Boston, Mass.

- Ehn P, (1988) *Work-Oriented Design of Computer Artifacts*, Arbetslivscentrum, Stockholm.
- Ekholm T, (1996) "Excelarket som blev IT-system", *Corporate Computing*, Nr 35, 28/8-1996.
- Engeström Y, (1991) "Developmental work research: Reconstructing expertise through expansive learning", I Nurminen M, Järvinen P & Weir G (red), *Proceedings of Human Jobs and Computer Interfaces Conference*, University of Tampere.
- Flensburg P, (1986) *Personlig databehandling*, Studentlitteratur, Lund.
- Fossum P, (1996) *Metoder för att hitta användbarhetsproblem hos datorsystem*, SISU Publikation 96:20, Svenska Institutet för Systemutveckling, Stockholm.
- Frenckner P, (1986) *Motiverat vägval vid avhandlingar i företagsekonomi*, W-serien 1986:1, ISSN 0349-5922.
- Gable G, Yap C S, Eng M N, (1991) *Spreadsheet Investment, Criticality, and Control*, Proceedings of the Twenty-Fourth Hawaii International Conference on System Sciences.
- Glaser B, (1978) *Theoretical Sensitivity*, The Sociology Press, California.
- Goldkuhl G, (1984) *Understanding computer-based information systems through communicative action analysis*, HUMOR, Institutionen för informationsbehandling, Göteborgs universitet.
- Goldkuhl G, (1990) *Aktörer i samarbete – Aktörsteori som grund för förståelse av förändringsarbete*, Institutionen för datavetenskap, Linköpings universitet.
- Goldkuhl G, (1998) *Kunskapande*, Institutionen för datavetenskap, Linköpings universitet.
- Goldkuhl G, (1993) *Verksamhetsutveckla datasystem*, Intention, Linköping.
- Goldkuhl G, (1995) "Information as Action and Communication", I Dahlbom B (red.): *The Infological Equation Essays in Honour of Börje Langefors*, Gothenburg Studies in Information Systems, Department of Informatics, Göteborg University.
- Goldkuhl G, (1996) *Informatik Ett ämne i, om och för förändring*, Föreläsning vid professorsinstallation, Internationella Handelshögskolan, Högskolan i Jönköping.
- Goldkuhl G (1996) *Praktikteoretiskt perspektiv som forskningsgrund*, Forskningsgruppen VITS, Institutionen för datavetenskap, Linköpings universitet.

Referenser

Goldkuhl G, Lyytinen K, (1982) *A Language Action View of Information Systems*, SYSLAB Report No 14, Stockholms universitet.

Goldkuhl G, Nilsson A, Röstlinger A, (1982) *Att specificera informationssystem - en användarorienterad och systematisk metodik*, Liber, Malmö.

Goldkuhl G, Röstlinger A, (1988) *Förändringsanalys – Arbetsmetodik och förhållningssätt för goda förändringsbeslut*, Studentlitteratur, Lund.

Goldkuhl G, Röstlinger A, (1997) *Praktikbegreppet – En praktikgenerisk modell som grund för teoriutveckling*, Forskningsgruppen VITS, Institutionen för datavetenskap, Linköpings universitet.

Goldkuhl G, Röstlinger A, (1998) *Praktikbegreppet – En praktikgenerisk modell som grund för teoriutveckling*, Forskningsgruppen VITS, Institutionen för datavetenskap, Linköpings universitet.

Greenbaum J, Kyng M, (red.) (1991) *Design at Work*, Lawrence Earlbaum, Hillsdale, N J.

Greve J, Karlsson B, (1994) *Att utöva inflytande med lokala informationssystem*, Licentiatavhandling, Företagsekonomiska institutionen, Uppsala Universitet.

Guimaraes T, Igarria M, (1997) "Assessing User Computing Effectiveness: An Integrated Model", *Journal of End User Computing*, Vol 9, No 2, Spring 1997, page 3-14.

Gummesson E, (1985) *Forskare och konsult - om aktionsforskning och fallstudier i företagsekonomi*, Studentlitteratur, Lund.

Gunton T, (1988) *Business Information Technology - End User Focus*, Prentice Hall, Hertfordshire.

Göranzon B, (1990) *Det praktiska intellektet*, Carlssons, Stockholm.

Habermas J, (1995) *Kommunikativt handlande Texter om språk, rationalitet och samhälle*, Daidalos, Göteborg.

Head R V, (1985) "Information Resource Center: A New Force in End-User Computing", *Journal of Systems Management*, Februari: sid. 24-29.

Henry J W, Martinko M J (1997) "An Attributional Analysis of the Rejection of Information Technology", *Journal of End User Computing*, Vol 9, No 4, Fall 1997, page 3-17.

Hellquist E, (1980) *Svensk etymologisk ordbok*, Gleerups, Malmö.

Hicks Jr J O, (1990) *Information Systems in Business: An Introduction*, West Publishing Company, St. Paul, MN.

Holmberg H, (1996) *Att välja rätt program för uppgiften - en jämförelse mellan Excel och Access*, C-uppsats, Institutionen för Ekonomi Statistik och ADB, Högskolan i Örebro.

Holme I M, Solvang B K, (1997) *Forskningsmetodik – Om kvalitativa och kvantitativa metoder*, Studentlitteratur, Lund.

Huber P H, (1990) "A theory of the Effects of Advanced Information Technologies on Organizational Design, Intelligence, and Decision Making", *Academy of Management Review*, 1990, Vol 15 Nr 1, sid 47-71.

Huff S L, Munro M, Martin B H, (1988) "Growth Stages of End User Computing", *Communications of the ACM*, Vol 31, Nr 5, Maj 1988, sid 542-550.

Hugosson M-Å, (1990) *Verksamhetsbaserad Systemstrukturering*, NordDATA90, Göteborg.

Isakovitz T, Schocken S, Lucas Jr. H C, (1995) "Toward a Logical/Physical Theory of Spreadsheet Modeling", *ACM Transactions on Information Systems*, Vol 13, Nr 1, Jan 1995, sid 1-37.

Isaksson A, (1996) *Excel för företagsekonomer*, Liber-Hermods, Malmö.

Johnson D G, (1985) *Computer Ethics*, Prentice Hall, Inglewood Cliffs, New Jersey.

Kaiser K M, Oppelland H J, (red.) (1990) *Desktop Information Technology*, Proceedings of the IFIP WIG 8.2 Working Conference on Desktop Information Technology and Organizational Worklife in the 1990's Ithaca, New York, USA, 2-4 July, 1989, North-Holland.

Keen P G W, (1991) "Relevance and Rigor In Information Systems Research: Improving Quality, Confidence, Cohesion and Impact", *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Elsevier Science Publishers B. V. (North-Holland).

Kjaer A, Madsen K H, (1995) "Dependencies between Work Activities, Technical Artifacts, Space and Work Organization: Flexibility issues", *Scandinavian Journal of Information Systems*, Vol 7(2): 23-44.

Kuhn T S, (1962) *The Structure of Scientific Revolutions*, University of Chicago Press, Chicago.

Referenser

- Lai V S, Mahapatra R K, (1997) "Exploring the research in information technology implementation", *Information & Management*, Nr 32 (1997), sid 187-201.
- Lally L, (1995) "Supporting Appropriate User-Developed Applications: Guidelines for managers", *Journal of End User Computing*, Vol 7, Nr 3, Summer 1995, sid 3-10.
- Langefors B, (1967) *Theoretical Analysis of Information Systems*, Studentlitteratur, Lund.
- Langefors B, (1986) "Information and Management Systems", *Erhvervsøkonomisk Tidsskrift*, 2'86, sid 135-148.
- Langefors B, (1993) *Essays on Infology*, Gothenburg Studies in Information Systems Report 5, University of Göteborg, Göteborg.
- Ledell G, (1993) *ADB-säkerhet i användargenomförd systemutveckling - en utmaning*, Dataföreningen i Sverige/Studentlitteratur, Lund.
- Lundahl U, Skärvad P, (1982) *Utredningsmetodik för samhällsvetare och ekonomer*, Studentlitteratur, Lund.
- Lyytinen K, Hirschheim R, (1988) "Information Systems as Rational Discourse An Application of Habermas's Theory of Communicative Action", *Scandinavian Journal of Management*, Vol 4, Nr 1-2, sid 19-30.
- Löwgren J, (1993) *Human-computer interaction What every system developer should know*, Studentlitteratur, Lund.
- Löwgren J, Stolterman E, (1998) *Design av informationsteknik*, Studentlitteratur, Lund.
- McLean E R, (1989) "End Users as Application Developers", In Nelson R R (ed.): *End-User Computing - Concepts, Issues, and Applications*, John Wiley & Sons.
- Mawhinney C H, Lederer A L, (1996) "An investigation of the Relationship Between Managerial Personality Type and Computer Use", *Journal of End User Computing*, Vol 8, No 1, Winter 1996, page 4-11.
- Mintzberg H, (1983) *Structure in fives: Designing effective organisations*, Prentice-Hall International, Englewood Cliffs, New Jersey.
- Molander B, (1988) *Vetenskapsfilosofi*, Thales, Stockholm.
- Molander B, (1996) *Kunskap i handling*, Daidalos, Göteborg.

- Moore J E, (1997) "The Application of Job Rotation in End User Computing: Toward a model for Research and Practice", *Journal of End User Computing*, Vol 9, No 1, Winter 1997, page 4-13.
- Munro M C, Huff S L, Marcolin B L, Compeau D R, (1997) "Understanding and measuring user competence", *Information & Management*, No 33 (1997), page 45-57.
- Mårtensson B, Nilstun T, (1988) *Praktisk vetenskapsteori*, Studentlitteratur, Lund.
- Mähring M, (1989) *Systemering vid utveckling av kalkylmodeller - Ett praktikfall och en metod för informationsstudie*, Uppsats, Handelshögskolan i Stockholm.
- Nardi B A, (1993) *A Small Matter of Programming*, The MIT Press, Cambridge, Massachusetts.
- Nardi B A, Miller J R, (1990) "An Ethnographic Study of Distributed Problem Solving in Spreadsheet Development", *Proc. ACM CSCW 90*. sid 197-208.
- Nationalencyklopedin (1993) Bokförlaget Bra Böcker, Höganäs.
- Nelson R R, (red.) (1989) *End-User Computing - Concepts, Issues, & Applications*, John Wiley & Sons.
- Nielsen J, (1993) *Usability Engineering*, Academic Press Ltd.
- Nilsson A G, (1987) "Systemering - en referensram", *Data* 4/87 sid 46-50.
- Nilsson A G, (1991) *Anskaffning av standardsystem för att utveckla verksamheter*, Doktorsavhandling, Handelshögskolan i Stockholm.
- Nissen H J, (1988) *The early History of the Ancient Near East 9000 - 2000 B.C.*, The University of Chicago Press, Chicago.
- Nordenstam T, (1985) *Technocratic and Humanistic Conceptions of Development*, Forskningsrapport 51, Arbetslivscentrum, Stockholm.
- Nordenstam T, (1993) *Från konst till vetenskap*, Carlssons, Stockholm.
- Norman J, (1993) *Kalkylprogrammet - ett personligt stödsystem?*, Examensarbete, Institutionen för datavetenskap, Universitetet i Linköping.
- Norman D A, (1993) *Things that make us smart Defending human attributes in the age of the machine*, Addison Wesley.
- Nurminen M, (1988) *People or Computers: Three Ways of Looking at Information Systems*, Studentlitteratur, Lund.

Referenser

- Nurminen M I, (1995) "Knowledge Work and Knowledge Act", I Dahlbom B (Red.): *The Infological Equation - Essays in Honor of Börje Langefors*, Report 6, Gothenburg Studies in Information Systems, Department of Computer Sciences, Chalmers University of Technology and the University of Göteborg.
- Olle T W, Hagelstein J, Macdonald I G, Rolland C, Sol H G, Van Assche F J M, Verrijn-Stewart A A, (1991) *Information System Methodologies*, Addison Wesley.
- Palmér J, Friedländer H, (Red.), (1964) *Ord för ord - Svenska synonymer och uttryck*. Nordiska uppslagsböcker, Stockholm.
- Patel R, Tebelius U, (Red.), (1987) *Grundbok i forskningsmetodik*. Studentlitteratur, Lund.
- Panko R R, (1988) *End User Computing - Management, Applications, & Technology*, John Wiley & Sons, New York, NY.
- Panko R R, Halverson Jr R P, (1996) *Spreadsheets on Trial: A Survey Research on Spreadsheet Risks*, Proceedings of the Twenty-ninth Hawaii International Conference on System Sciences.
- Panko R R, (1996) *Hitting the Wall: Errors in Developing and Debugging a "Simple" Spreadsheet model*, Proceedings of the Twenty-ninth Hawaii International Conference on System Sciences.
- Patel R, Tebelius U, (Red.) (1987) *Grundbok i forskningsmetodik*. Studentlitteratur, Lund.
- Patton M Q, (1980) *Qualitative Evaluation Methods*. Sage, London.
- Phillips E M, Pugh D S, (1994) *How to get a Ph.D. - A handbook for students and their supervisors*, Open University Press, Buckingham, Philadelphia.
- Repstad P, (1987) *Närhet och distans - Kvalitativa metoder i samhällsvetenskap*, Studentlitteratur, Lund.
- Reynolds G W, (1992) *Information systems for managers*, West Publishing Company, St Paul, MN.
- Rivard S, Huff S L, (1988) "Factors of Success for End-User Computing", *Communications of the ACM*, Vol 31, Nr 5, May 1988, sid 552-561.
- Rockart J F, Flannery L S, (1989) "The management of End-User Computing", In Nelson R R, (Red.): *End-User Computing - Concepts, Issues, and Applications*, John Wiley & Sons.

Rolf B, (1978) *Profession, tradition och tyst kunskap*, Nya Doxa, Övre Dalkarlshyttan.

Rosing H, (1978) *Vetenskapens logiska grunder*, Schildts, Åbo.

SAO (1973) *Svenska Akademiens Ordlista över svenska språket*, P A Norstedt & söners förlag, Stockholm.

Saldarini R A, (1989) *Analysis and Design of Business Information Systems*, Macmillan Publishing Company, New York, New York.

Schön D A, (1982) *The Reflective Practitioner*, BasicBooks, New York.

Searle J R, (1969) *Speech acts. An essay in the philosophy of language*, Cambridge University Press, London.

Shah H U, Lawrence D R, (1996) "A study of End User Computing and the Provision of Tool Support to Advance End User Empowerment", *Journal of End User Computing*, Vol 8, No 1, Winter 1996, page 13-21.

Smithson S, Hirschheim R, (1990) "End-user computing and the user-system interface: A five level framework", In *Desktop Information Technology*, North Holland, Amsterdam, sid 37-57.

Speier C, Brown C V, (1997) "Differences in end-user computing support and control across user departments", *Information & Management*, Nr 32 (1997), sid 85-99.

Sprague R H, Watson H J, (1996) *Decision Support for Management*, Prentice-Hall.

Stabell C B, (1987) "Decision Support Systems : Alternative Perspectives and Schools", *Decision Support Systems (3)* 243-251, Elsevier Science Publishers B. V. (North-Holland).

Stabell C B, (1988) *A Decision Research-Based Approach to DSS Development*, Working Paper, Norwegian School of Management, Norge.

Starrin B, Larsson G, Dahlgren L, Styrborn S, (1991) *Från upptäckt till presentation - Om kvalitativ metod och teorigenerering på empirisk grund*, Studentlitteratur, Lund.

Storey V C, Chan Y E, (1994) *The Use of Spreadsheets in Organizations: Determinants and Consequences*, Queen's University School of Business Research Program, Kingston, Ontario.

Svenning C, (1996) *Metodboken*, Lorentz Förlag.

Referenser

Sundgren B, (1992) *Databasorienterad systemutveckling - grundläggande begrepp, datamodellering, systemkonstruktion*, Studentlitteratur, Lund.

Tafti M H A, (1997) "Hierarchy of End User Computing Needs: An Empirical Investigation", *Journal of End User Computing*, Vol 9, No 4, Fall 1997, page 29-35.

Tempte T, (1997) *Lilla arbetets ära*, Carlssons, Stockholm.

Turban E, (1992) *Expert Systems and Applied Artificial Intelligence*, MacMillan, New York, N Y.

Wallén G, (1996) *Vetenskapsteori och forskningsmetodik*. Studentlitteratur, Lund.

Walters J R, Nielsen N R, (1988) *Crafting Knowledge Based Systems*, John Wiley & Sons, New York.

Waterman D A, (1985) *Guide to Expert Systems*, Addison-Wesley, Reading, Mass.

Watson H J, Houdeshel G, Rainer Jr R K, (1997) *Building Executive Information Systems and other Decision Support Applications*, John Wiley & Sons.

Westin O, (1993) *Informationsstöd för lokal ekonomi - En studie kring centrala informationssystem och lokala informationsstöd ur ett verksamhetsperspektiv*, BAS, Göteborg.

Wilkins M L, Nantz K S, (1995) "Faculty Use of Electronic Communications Before and After a LAN Installation: A Three-Year Analysis", *Journal of End User Computing*, Vol 7, No 1, Winter 1995, page 4-11.

Wilkinson L, (1989) *SYSGRAPH: The system for graphics*, Evinston IL.

Weinberg G M, (1971) *The Psychology of Computer Programming*, Van Nostrand Reinhold Company, New York.

Westerlund F, (1997) *Kunskapen om kalkylprogram som faktor för lyckade kalkylsystem*, Examensarbete, Institutionen för ekonomi, statistik och ADB, Högskolan i Örebro.

Winograd T, Flores F, (1993) *Understanding Computers and Cognition*, Addison-Wesley, Norwood, NJ.

Yourdon E, (1989) *Modern Structured Analysis*, Prentice-Hall, Englewood Cliffs, NJ.

Övriga källor

Carlsson S, (1997) Föreläsningssanteckningar från doktorandkursen *Decision Support for Management*, Institutionen för datavetenskap, Linköpings universitet vt 1997.

Index

4

4GL-verktyg, 181

A

abduktion, 48
 ACM, 22, 263, 264, 269
administration av tomrättskontrakt, **102**
 Agnér Sigbo, 201
 AIS-K, 13
 aktualitet, 67, 75, 301, 303, 305, 322
 aktör, 217, 234
allokering av restpallar, **68**
 Alter, 203, 206, 207, 210, 213
alternativa hyreskostnader, **97**
 Alvesson & Sköldberg, 36, 39, 49
 AMIS-projektet, 201
 analys, 82
 Andersen, 160, 162, 179, 181, 182, 183, 192, 195, 199, 200, 203, 209, 210, 213, 218, 275
 Angehrn, 235
 anonymisering, 56
 ansvar, 16, 149, 270
 ANTAL.OM, 328
 anvecklade informationssystem, 166
 anvecklare, 15, 17, 220, **224**
 anveklargeneriska modellen, 333
 anveklarpraxis, 212
 anveckling, 23, **24**, 189, 190
 användare, 9, 217, 220
 användarmedverkan, 177
 användarprototyping, 186, 298
 användarutvecklare, 220
 användbarhet, **283**
 användbarhet, utvecklingsverktyg, 287
 användning, 14
 Application development, 274
 Applikation, 131, **169**, 298
 Applikationen, 65, 78, 84, 86, 95, 132, 186, 210, 318
 Approach, 120
 arbetsfördelning, 208
 arbetsintegrerad systemutveckling med kalkylprogram, 13
 arbetsuppgiftsperspektiv, 192
 Argyris & Schön, 254, 271
 artificiell intelligens, 22, 257
 AssiDomän Frövi, 61, 122
 a-system, 153, **166**, 167
 attityd, 301
 automatisering, 6
 Avdic, 21, 40, 43, 54, 84, 152, 167, 172, 184, 209, 214, 225, 230, 242, 276
 avsaknad av standardiserat metodarbete, **194**
 avveckling, 182
 avvikelser, 86
 Axelsson & Ortman, 181

B

Bansler, 161, 179
 Basic Use Skills, 241
 Beekman, 181, 229, 275
 begrepp, 36
 begreppsgrafer, 53
 Benham, Delaney & Luzy, 210
 Berg & Hultman, 181
 Berger & Luckman, 165, 251, 252
beräkning av arkantal per pall, **70**, 164, 177, 243
beräkning av cash-flow, **97**
beräkning av fastighetsskatt, **108**
beräkning av kreditivräntor, **108**, 231
 beräkning/simulering, 320
 berättigande, 197
 beskrivande strategi, 34
 beskrivningar, 183
 beslutsstöd, 128
 beslutsstödande syfte, **226**, 304
 beslutsstödsystem, 198, 228
 blockerande förförståelse, 56
 Bogdan & Taylor, 37, 55
bokslutssystem, **108**, 179, 199, 298, 304
 Borland Paradox, 194
 Brancheau & Brown, 191, 194, 196, 213, 214
 Bricklin, 276
 Bubenko jr & Lindencrona, 240
 Budde, 181
budgetering av exploateringsföretag, **95**, 168
budgetsystem, **106**
budgetuppföljning, **184**

C

C, 181
 C++, 186
 CAD-program, 276
 CalcResult, 94
 Cameron, 181
 Capron, 210, 229
 Carlsson, 169, 191, 198, 203, 206, 235, 275
 cash-flow, 97
 cellskydd, 73
 Cobol, 181, 186, 302
 Collste, 21, 261
 Comfortable Use Skills, 242
 Computer Supported Work, 257
 Conger, 191
 Courbon, 235

D

Dahlbom & Mathiassen, 240, 244, 245, 259
 Data management, 274
 Data Mining, 119, 307
 Data Warehouse, 119, 125, 307
 dataaccess, 234
 databashanteringsprogram, 128

Användare och utvecklare

dataspecialist. *Se* IT-specialist
datastrukturering, 319
datorvana, 137
Davenport, 8
DDT, 214
Decision Support Systems. *Se* beslutsstödssystem
deduktion, 48
den detaljerade tumregelsansatsen, 214
detaljeringsgrad, 74
dimensionering av bropelare, **102**, 304
direktiv information, 227
distans, 55
distribuerad budget, **131**, 304
dokumentation, 121, 126, 203
domän. *Se* kunskapsområde
double-loop learning, 271
DSS, 230, 234. *Se* beslutsstödssystem
DSS-generator, 234
dubbellagring, 205
dynamisk verksamhet, 303
Dyno Nobel, 118

E

effectiveness, 232
effekter KPA, 26
Ehn, 274
EIS, 229
ekonomiska effekter, 205
empirinära metod, 51
enabler, 8
End User Computing, 157, **189**, 230
End User Computing, definition, 191
ENIAC, 4
enkelhet, 243, 319
erfarenhet, 156, 190
erfarenheter, 148
ersättning, 147
EUC, 189. *Se* End User Computing
EUC och anveckling, 190
Executive Information Systems, 229
exekutivt effektiv information, 227
exempel, 253
expansion, 196
explorativ strategi, 34

F

fadderanveckling, 79, 221, 283
fallstudie, 49
FAR, 22, 269
FEL-funktioner, 328
felhantering, 289
fenomenell aspekt, 313
filtyper, 325
Flensburg, 182, 192, 210, 213
flexibilitet, 243, 319
Floyd, Walls & Marr, 169
fokal dimension, 250
formalisering, **6**, 81, 150, 245, 312, 323
formelkonstruktion, 323
formler, 126
forskningsfråga, **27**
forskningsstrategi, 34
Fossum, 285
Four Strategies for EUC Growth, 195
frågeverktyg, 322

funktionalitet, 284
funktionalitets befintlighet, 288
funktionalitets tillgänglighet, 288
funktionell aspekt, 313
funktioner, 323
fördelning av arkning, **71**, 169
förförståelse, 41, 42
förkategorisering, 177
förklarande strategi, 35
förmedlare, 219
Förnyelsesynsätt, 214
förståelse, 41
förståelsekunskap, 240, 258
förvaltning och drift, 182
föränderlighet, 67
förändring, 83, 303
förändringsanalys, 182
förändringsanalys, avsaknad, 203
förändringsanalys, implicit, 210

G

G(D)SS. *Se* gruppsystem
Gable, Yap & Eng, 205
GDSS. *Se* gruppsystem
GIS, 109
Gogan, 192
Goldkuhl, 34, 36, 41, 160, 162, 164, 179, 219, 221, 239, 240, 259
Goldkuhl & Lyytinen, 165
Goldkuhl & Röstlinger, 30, 145, 262
Goldkuhl, Nilsson & Röstlinger, 10
Good Practice, 242
grafisk analys, 323
granskning, 281, 325, 326, 328
Greve & Karlsson, 194, 207, 208, 276
grupp(besluts)system. *Se* gruppsystem
grupp(utveckling), 221
gruppsystem, 229
guider, 274
Gummesson, 55
Göranzon, 258, 267, 309

H

habitualisering, 251
handling, HIS, 171
handlingsansvariga, 219
hands-on, 73
hantering, 323
Helenius, 41
hermeneutik, 41, 51
hermeneutiska spiralen, 41
Hicks Jr, 23, 201, 203, 207, 211, 214
Hippokrates, 266
HIS, **169**
hjälpmedel, 274
Holmberg, 194, 234, 275, 276
Holme & Solvang, 37, 38
Huber, 232, 306
Human-scale Information System, 170
hybridmetod, 298
hypotestestande strategi, 35
Hägg, Wiedersheim-Paul & Aronsson, 41
hög-kontroll-synsättet, **208**, 267, 299

Index

I

icke-linjärt arbetssätt, 54
icke-procedurell utveckling, **194**
ifrågasättande, 336
import/export, 320
inavvändare, 219
INDEX, 84, 319, 320, 327
individuellt lärande, 206
induktion, 48
induktivt arbetssätt, 210
infologiska ekvationen, 9
Information Center, 210, 213
informationskvalitet, 204
informationssystem, 159, **160**, 161
informationssystemanalys, 182
informationssystemutveckling, 57
informella nätverk, 196
införande, 182
Innovation, 242
inre validitet, 56
institutionalisering, 251
instrument, 147
integration, 194, 278, 299, 335
intentioner, 147
interaktiv kontroll, 81
interaktivitet, **13**, 137, 234, 278, 299, 335
Internet, 320
intranät, 84
investering per verksamhet, **107**
Isakovitz, Shocken & Lucas Jr, 169, 192, 211, 214, 276
Isaksson, 214
IT, 3
IT som möjliggörare, 242, 300, 307
IT-specialist, 9, 15, 217, 219, 308
IT-specialistperspektiv, 117

J

Johnson, 21

K

Kaiser & Oppelland, 168, 192
kalkylblad, 277
kalkylprogram, **13**, 42, 234, **276**, 277, 286
kalkylprogram, funktioner, 278
kalkylprogramanveckling, **13**, 23, 192
kalkylsystem, **167**, 230, 234
kapitalvärdesutveckling, 97
Karlsson, 298
kataloger, 326
Klepper, 205
Kling, Iacono & George, 206
know-how, 253, 264
kodning, 51
kommunicerbar information, 246
kommunicerbarhet, 47
kommunikation, 79, 319, 325
kommunikation, HIS, 172
kommunikativa funktioner, 164
kompetens, 254, 264
konkretisering, 177
kontinuerlig förändring, 334
kontinuerliga förändringscirkeln, 83
kontroll, 70, 196, 324
kontroll och underhåll, 204

kontrollberäkningar, 80
koordination, 208
kostnadsjämförelse för arkning, **74**, 268, 270, 324
KPA, 13, 229
KPA Livscykelmodellen, 184
KPA, arbetssätt, 73, 84, 102, 110, 114
KPA, effekter, 26, 110
KPA, problem, 72, 84, 102, 110, 119, 121, 124, 125, 127
kunnande, 147
kunskap, **239**, 307, 309
kunskap om utvecklingsverktyg, 17, 289
kunskap, HIS, 170
kunskap, objektivering av, 311, 314
kunskaper om kalkylprogram, 317
kunskapsbehov, 128
kunskapsbildningsprocessen, 47
kunskapsområde, 151
kunskapsstyper, 138, 307
kunskapsöverföringsproblemet, 10
kvalitativa fel, 280
kvantitativa fel, 280
Kör webbfråga, 307

L

Lally, 23, 191, 203
Langefors, 9, 161, 168, 181, 195
layoutprogram, 276
Ledell, 13, 169, 280
ledning, 128
LETARAD, 320
LETAUPP, 126
leverantören, 224
Lilla kalkylsystemet, 65, 78, 95, **168**, 318
Livscykelmodellen, **180**, 181, 184, 195, 210, 212, 292, 300
Ljungdahl, 41
logisk kontroll, 324
logiska fel, 281
lokala databaser, 306
lokala nätverk, 80, 235, 302, 315, 325
lokalt informationsstöd, 207
Lotus 1-2-3, 276
Lundahl & Skärvad, 34, 35, 42
Lyytinen & Hirschheim, 164
låg-kontroll-synsättet, **209**, 267, 299
lämplighet, 197
länkning, 84, 126, 320, 322
Löwgren & Stolterman, 159

M

makro, 126
makroinspelning, 323, 327
mallar, 327
management, 229
Management Information System, 228
markvärdering, **102**, 268
maskin, 275
mekaniska fel, 281
mental modell, 285
metod, 274
metoder, avsaknad av standardiserade, 210
metodtriangulering, 40
miljö, 186
Mintzberg, 208
MIS, 302

Användare och utvecklare

Molander, 50, 240, 252, 255, 268, 271
Moore, 12
MS Access, 109, 128, 181, 194
MS Excel, 62, 72, 96, 109, 123, 124, 126, 128, 231, 234, 236, 247, 258, 263, 276, 281, 307, 310, 320, 325, 329
MS Power Point, 123
MS Query, 307, 310, 320
MS Word, 109, 123
multimetodprincipen, 52
multisinnesprincipen, 52
mål, 263
mål, många föränderliga, 301
målbaserad praktik, 149, 266
målbaserade normer, 266
målbaserat arbete, 233
målgrupp, 29
målstyrning, 301, 307
många mål, 75
Mårtensson & Nilstun, 36, 41
Mähring, 210, 212
människan, HIS, 171
möjliggörare, 8, 45
möjlighet, 197

N

namn, 328
Nardi & Miller, 13
Nationalencyklopedin, 6, 13, 226, 263
Nelson, 168
Nielsen, 284
Nilsson, 24, 177, 181, 224
Nissen, 291
noggrannhet, 47, 66, 89, 103, 301, 303, 305, 322
Nordenstam, 240, 256
normer, 147, **262**, 263
normer, omprövning av, 303
Nurminen, 43, 45, 161, 169, 209, 227, 240, 247, 257, 274, 297, 306
närhet, 55

O

oberoende, strävan efter, 210
Ogdens triangel, 244
OM, 72, 320, 327, 328
ontologisk aspekt, 313
OOA, 302
operativ information, 227
Ordbehandlarsystemet, 78, 95, **168**, 318
ordbehandling, 53
organisation, HIS, 172
organisationsexterna databaser, 306
organisationsinterna databaser, 306
Organisationssynsätt, 213
ostrukturerade länkar, 84
osäkra indata, 281

P

Panko, 13, 168, 210, 211, 241, 274, 277
Panko & Halverson Jr, 169, 212, 280
Parker, Verner & Cerpa, 191, 195
Pascal, 186
Patel & Tebelius, 34, 36, 37, 41
Patton, 37
pedagogiskt syfte, 71

permanenta kalkylsystem, 80
personlig databehandling, 213
personlig kunskap, 258
perspektivväxling, 43
pivottabeller, 309
Polanyi, 248, 249, 252
potentiellt beslutsstödjande syfte, 304
power users, 212
pragmatisk kunskapssyn, 240
praktik, **146**
praktikgeneriska modellen, **145**
praktikkunskap, 241
praktisk kunskap, 258
presentation, **73**, 86, 89, 120, 123, 128, 319
presentation av förbrukning, **101**
principen om estetisk distans, 52
pris- och indexutveckling, **100**, 163, 177, 231, 329
Problemlösaren, 247, 323
problemområde, behandling av, 46
problemområde, val av, 46
procedurbaserad praktik, 149, 266
procedurbaserade normer, 267
procedurbaserat arbete, 233
producenten, 148
produktionsuppföljning, **69**, 305, 312
produktivitet, 206
produktkalkyl, **80**, 243
profession, **254**
professionskunskap, 308
programmering, 319, 322, 323, 327
prototyping, 124, 181, 298
påståendekunskap, 240, 258

Q

Quattro Pro, 42, 276

R

radikal konventionalism, 248
rationaliserande syfte, **226**, 304
realisering, 182
redigering, 322, 323
redovisning av kostnader, **109**
reifiering, 251
relevans, 103
Repstad, 55
resultat, 148
resultatberäkningssystem, **184**
resursslöseri, 205
revidering, 314
Reynolds, 199, 202, 203, 204, 229, 275
rimlighetsbedömning, 69, 80, 303, 305, 324
Rivard & Huff, 200, 275
Robey & Zmud, 197, 198, 206, 213
Rolf, 248, 250, 252, 253, 264, 299, 313
Rosing, 36, 274
rubrikläsning, 328

S

sammanflätade aktiviteter, **193**, 210, 299
sammanställning, 95, 320, 323
sammanställning av en personalenkät, **112**, 169, 233
Schön, 254, 299
Searle, 165
selektiv perception, 55

Index

semantisk aspekt, 313
Séror, 206
simulering, 234, 323
SIS-RAS, 302
självförsörjning, 221
skill/skicklighet, 253
skärmdelning, 328
slutliga mottagare, 219
Smithson & Hirschheim, 206, 207, 209, 285, 287
socialfenomenologi, 165
specialfunktioner i verktyg, 319
Sprague & Watson, 149, 229, 230, 233, 235
spreadsheet program, 13
SQL-frågor, 62, 113, 124, 310, 325
Stabell, 324
Stadsbyggnadskontoret i Örebro, 91
standardsystem, 181, 224
Starrin m.fl., 51
statistik över bygglovsärenden, **113**, 311
statistikprogram, 276
stegvis förfining, 176, **193**, 209
Stora kalkylsystemet, 78, 84, 95, **169**, 212, 318
Storey & Chan, 169, 191
strategiska mål, 82
strategiskt syfte, **226**, 303, 304
strukturella fel, 281
styrning, 204, 207, 208
SUMMA, 243
SUMMA.OM, 328
Sundgren, 160, 180, 218, 240
superanvecklare, 16
Svenning, 36, 40, 56
svårformaliserbar kunskap, 22, 245
svårformaliserbara mål, 261
syfte avhandling, 28
syfte med kalkylsystem, 304
syften med kalkylsystem, 225
systemarbete, HIS, 192
systemering, 182
systemintegration, 204
systemutveckling, **175**
systemutvecklingsperspektiv, 193

T

talaktsteori, 165
taraberäkning, **73**, 328
teknologinivåer, 231
Tempte, 269, 270
teoretisk känslighet, 52
teorigenerering på empirisk grund, 51
theory-in-use, 271
tidrapport, **96**, 169, 177
tillfälliga kalkylsystem, 80
tillgänglig kunskap, 6
TIS, **166**, 289
tolkning, 41
tomtprisberäkning, **101**
tomtprisberäkningssystem, **98**, 268, 304
Traditionalismsynsätt, 211
traditionell systemutveckling. *Se* TSU
traditionella informationssystem. *Se* TIS
Triple C, 235
TSU, 10, 176, **178**, 180, 209, 283, 310
tunnelseende, 300
tyst dimension, 250
tyst kunskap, 6, 23, **247**, 311, 312

tyst kunskap av andra ordningen, 253
täckningsbidrag per timme, **82**, 169, 184, 203, 230, 233,
259, 271, 304, 312

U

underlag, 147, 306
uppdrag, 147
uppföljning av arkmaskiner, 252, 268, 271
uppföljning av arkningsmaskiner, **87**, 233, 298, 304, 305,
311
uppgiftstyp, 234
ursprungliga formulerare, 219
utanvändare, 219
utbildning, 196, 329
utelämnande fel, 281
utformning, 182
utförandekunskap, 240, 258
utvecklare, 217
utveckling, 14
utvecklingsverktyg, 17
utvecklingsverktygskunskap, 308

V

validitet, 56
Wallén, 34, 35, 40, 48
Waterman, 197
Watson m.fl., 229, 230
vattenfallsmodellen, 180
webbanveckling, 338
webbverktyg, 322
verifierande syfte. *Se* beslutsstödjande syfte
verifiering, 70, 305, 314
verksamhet, 9
verksamhetsanalys, 182
verksamhetsbehovsperspektiv, 45
verksamhetsetik, **22**, 262
verksamhetsföreträdare, 9, 217
verksamhetsklienter, 219
verksamhetskunskap, 18, 308
verksamhetsmotverkande chauvinism, **19**, 203, 254, 300
verktyg, **273**, 274
verktygskunskapsperspektiv, 45
versionsbyte, 121, 128, 204, 205, 329
Westerlund, 243, 319
Westin, 169, 207
what-if, 42, 234
villkorsformler, 328
VisiCalc, 276
Visual Basic, 181
visualisering, 53

Y

Yourdon, 180, 181, 218
yrkes- och verksamhetsetik, **22**
yrkesetik, **21**, 262
yttre validitet, 56

Z

Ziguraten i Ur, 291

Användare och utvecklare

Å
återanvändning, 327

Ä
ändamålsenlighet, 243, 319

ändringsbarhet, 72
ÄRFEL, 328

Ö
Ödman, 41
ömsesidig anpassning, 208

Bilagor

Bilaga 1: Watermans modell

Bilaga 2: Intervjufrågor kalkylprogramanvecklare

Bilaga 3: Intervjufrågor IT-specialister

Bilaga 4: Intervjufrågor användare, intervjufrågor systemägare

Bilaga 5: ACM's 'Code of Professional Conduct

Bilaga 6: Etiska regler för kommunala ekonomer

Bilaga 1

Nedanstående uppställning (Turban 1993:407ff) beskriver Watermans modell.

Requirements for ES Development. The following twelve requirements are *all* necessary to make ES development successful:

1. The task does not require common sense.
2. The task does only require cognitive, not physical, skills.
3. At least one genuine expert, who is willing to cooperate, exists.
4. Experts involved can articulate their methods of problem solving.
5. Experts involved can agree on the knowledge and the solution approach to the problem.
6. The task is not too difficult.
7. The task is well understood and is defined clearly.
8. The task definition is fairly stable.
9. Conventional (algorithmic) computer solution techniques are not satisfactory.
10. Incorrect or non-optimal results can be tolerated.
11. Data and test cases are available.
12. The task's vocabulary has no more than a couple of hundred concepts.

Justification for ES Development. Like any other information system, an expert system needs to be justified. Of the following eight factors, at least one must be present to justify an ES:

1. The solution to the problem has a high payoff.
2. The ES can preserve scarce human expertise so it will not be lost.
3. Expertise is needed in many locations.
4. Expertise is needed in hostile or hazardous environments.
5. The expertise improves performance and/or quality.
6. The system can be used for training.
7. The ES solution can be derived faster than that which a human can provide.
8. The ES is more consistent and/or accurate than a human.

The derived benefits in one or more of these areas must be compared against the costs of developing the system. A preliminary justification is conducted in this phase, whereas a detailed analysis is performed in phase II.

Appropriateness of the AS. Waterman suggests that the following three factors be considered in determining when it is appropriate to develop an ES.

1. Nature of the problem: The problem should have a symbolic structure, and heuristics should be available for its solution. In addition, it is desirable that the task be decomposable.
2. Complexity of the task: The task should be neither too easy nor too difficult for a human expert.
3. Scope of the problem: The problem should be of manageable size, it also should have some practical value.

Bilaga 2

Intervjufrågor kalkylprogramanvecklare

Företag?

Verksamhet?

Titel?

Funktion?

Mål med Ditt arbete?

Användningsfrekvens kalkylprogram?

Övriga programvaror som Du använder?

I vilken datormiljö verkar Du?

Hur länge har Du använt kalkylprogram?

Vad använder Du kalkylprogram till?

Hur gör Du när Du utvecklar kalkylsystem?

Kan Du göra saker med kalkylprogram som Du inte skulle kunna ha gjort annars?

Kan Du ta hänsyn till övergripande mål?

Problem med användning av kalkylprogram (KPA)?

Lösningar på problem?

Alternativ till användning av kalkylprogram (KPA)?

Dokumenterar Du Dina kalkylsystem? (Om inte, varför?)

Förebygger Du förändringar som kan påverka kalkylsystemen?

Exempel på kalkylsystem?

Exempel på lyckat kalkylsystem?

Bilaga 3
Intervjufrågor IT-specialister

Företag?

Verksamhet?

Titel?

Funktion?

Mål med Ditt arbete?

Egen användningsfrekvens kalkylprogram?

Övriga programvaror som används?

I vilken datormiljö verkar Du?

Problem med användning av kalkylprogram (KPA)?

Vilka lösningar finns på problemen?

Vilka alternativ finns till användning av kalkylprogram (KPA)?

Dokumenteras kalkylsystem? (Om inte, varför?)

Hur förebyggs förändringar som kan påverka kalkylsystemen bäst?

Vilka fördelar finns med användning av kalkylprogram (KPA)?

Exempel på kalkylsystem?

Exempel på lyckat kalkylsystem?

Bilaga 4

Intervjufrågor användare

Företag, avdelning?

Verksamhet?

Titel?

Funktion?

Mål med arbetet?

Hur upplevde Du budgetering via systemet?

Vad var bra?

Vad var dåligt?

Budgeterade Du interaktivt?

Blev resultatet bättre?

Intervjufrågor systemägare

Mål med systemet?

Intressenter?

Systemfunktioner OK?

Antal användare?

Användningsfrekvens bland användarna?

Olika användningsgrupper?

Äldre-yngre?

Sulfat-KM5 ? etc.

Hanteringsproblem?

Skedde budgeteringen via bildskärm?

Utnyttjades möjligheterna?

Var det några som utnyttjade möjligheterna till självständig analys?

Var det problem med flera kalkylblad?

Bilagor

Spelar det roll papper-bildskärm?

Utvecklingsarbetet? Beställarens deltagande, hur upplevdes detta? Kontroll?

Skulle Du föredragit att lägga ut det som ett jobb till dataavdelningen?

Rapporteringssystemet, har det fungerat bättre?

Användarutbildning, erfarenheter?

Vad var det som gjorde att det fungerade? Egenskap hos användare, etc.?

Vad var det som gjorde att det inte fungerade?

Personliga reflexioner?

Bilaga 5

The American organization 'Association for Computer Machinery' (ACM) has put together a 'Code of Professional Conduct' for its members. The Code stipulates how a computer professional should act on different occasions and what norms there are when working with computers.

Canon 1

An ACM-member shall act at all times with integrity.

Ethical considerations:

1.1 An ACM-member shall properly qualify himself when expressing an opinion outside his areas of competence.

1.2 An ACM-member shall preface any partisan statements about information processing by indicating clearly on whose behalf they are made.

1.3 An ACM-member shall act faithfully on behalf of his employers or clients.

Canon 2

An ACM-member should strive to increase his competence and the competence and prestige of the profession.

Ethical considerations:

2.1 An ACM-member is encouraged to extend public knowledge, understanding and appreciation of information processing.

2.2 An ACM-member shall not use his professional credentials to misrepresent his competence.

2.3 An ACM-member shall undertake only those professional assignments and commitments for which he is qualified.

2.4 An ACM-member shall strive to design and develop systems that adequately perform the intended functions and that satisfy his employer's or client's operational needs.

2.5 An ACM-member should maintain and increase his competence through a program of continuing education encompassing the techniques, technical standards, and practices in his fields of professional activity.

2.6 An ACM-member should provide opportunity and encouragement for professional development and advancement of both professionals and those aspiring to become professionals.

Canon 3

An ACM-member shall accept responsibility for his work.

Ethical considerations:

3.1 An ACM-member shall accept only those assignments for which there is reasonable expectancy of meeting requirements or specifications, and shall perform his assignments in a professional manner.

Canon 4

An ACM-member shall act with professional responsibility.

Ethical considerations:

4.1 An ACM-member shall not use his membership in ACM improperly for professional advantage or to misrepresent the authority of his statements.

4.2 An ACM-member shall conduct professional activities on a high plane.

4.3 An ACM-member is encouraged to uphold and improve the professional standards of the Association through participation in their formulation, establishment and enforcement.

Canon 5

An ACM-member should use his special knowledge and skills for the advancements of human welfare.

Ethical considerations:

5.1 An ACM-member should consider the health, privacy and general welfare of the public in the performance of his work.

5.2 An ACM-member, whenever dealing with data concerning individuals, shall always consider the principle of the individual's privacy and seek the following:

- To minimize the data collected.
- To limit authorized access to the data.
- To provide proper security for the data.
- To determine the required retention period of the data.
- To ensure proper disposal of the data.

There are a number of 'disciplinary rules' attached to each canon. These rules formulate more concretely how an ACM-member shall or shall not act in a certain situation.

In general one could say that a computer professional acts ethically when he:

- is faithful to his employers and clients/customers
- is honest
- increases his own and the profession's competence
- strives for quality
- accepts responsibility for his work and his profession
- cares for human welfare
- considers privacy.

Comments on the ACM-canon.

The rules describe how a computer professional should act. Another formulation of this could be as follows.

A computer professional shall not pretend to a greater knowledge than he actually has.

A computer professional shall be honest and declare on whose behalf he is acting.

A computer professional shall keep his promises.

A computer professional shall increase the prestige of the profession.

A computer professional shall only accept jobs he can manage.

A computer professional shall try to fulfil the real needs of his employers and clients.

A common characteristic for the ethical rules is that they concern the *relations* between computer professionals and their employers and clients/customers.

Bilaga 6

Etik för kommunalekonomer

Tretton etiska regler inom Föreningen Sveriges Kommunalekonomer

Med något förkortade formuleringar.

1. Visa respekt för och uppträd så Du stärker förtroendet för kommunal verksamhet
2. Verka för god ekonomisk hushållning och för att ekonomiska aspekter beaktas i verksamheten
3. Arbeta aktivt med att klargöra Din roll och spelreglerna inom ekonomiadministrationen
4. Grip varje tillfälle för att öka kunskapen om och förståelsen för ekonomin
5. Agera alltid professionellt och utveckla ständigt Ditt yrkeskunnande
6. Dölj aldrig relevanta fakta om kommunens ekonomi och ta de initiativ som krävs för att bevara en god ekonomisk hushållning
7. Verka för att beslut fattas efter saklig beredning och i medvetande om de ekonomiska konsekvenserna
8. Var opartisk i relation till alla kommunmedborgare och alla förtroendemän i kommunen
9. Värna om en öppen och korrekt information
10. Missbruka inte uppgifter som Du har fått i förtroende
11. Tänk på att Du förvaltar allmänna medel. Spekulera inte, ta inga onödiga risker
12. Uppträd så att Du väcker respekt för Dig själv och den yrkeskår Du representerar
13. Kommunalekonomer stödjer varandra aktivt för att upprätthålla en god yrkesetik