

KANDIDATUPPSATS I INFORMATIK  
VID INSTITUTIONEN FÖR DATA- OCH AFFÄRSVETENSKAP  
1997:K23

# Användarmedverkan vid systemutveckling

Johan Sanneblad

Handledare: Bertil Lind

vt 1997



HÖGSKOLAN  
I BORÅS



# Abstract

In modern society information technology is becoming more or less a standard in every organization. The purpose of this report is to clarify the ongoing discussion that exist on making systems development a more humanistic procedure, where the employees participate freely during most of the stages included in the process. In order to achieve this, three different development methods have been investigated where each method uses a slightly different approach to its users. The results illustrate that there has been a recent change in the approach to user participation in systems development, mostly dependent on the major advances in object oriented development systems that has become available. These tools and methods do not use a linear development path, but instead several developers are able to work simultaneously on the same project, in different phases. This means that even if the users participate in the analysis phase only, it does not necessary have to be a one-time opportunity.

**Nyckelord:** Användardeltagande, användarmedverkan, datasystem, informationssystem, systemutveckling, verksamhetsmodellering



# Förord

Då en studie av något så dagsaktuellt som systemutvecklingsmetoder hade varit omöjlig att genomföra utan kontakt med näringslivet, så måste jag säga att bemötandet varit mycket mer positivt än jag i inledningsskedet vågat hoppats på. De personer jag framförallt vill tacka är Inger Boivie på Enator och Pär Carlshamre på WM-Data, som båda bidragit med sina respektive företags metoder till studien. Boivie skall även ha en eloge för att hon tog sig tid att granska de idéer jag haft under skrivandets gång, och bidragit med ett flertal litteraturtips och kommentarer.

Även inom den akademiska världen har bemötandet varit mycket positivt, och jag vill särskilt passa på att tacka Kerstin Grundén för värdefulla kommentarer tillsammans med Bertil Lind som bidragit med ett flertal kommentarer, litteraturtips och idéer.

Arbetet inom ITCentrum har bidragit till att mitt intresse för användardeltagande åter väckts, och för denna möjlighet vill jag tacka centrets föreståndare Per Zaring, som även han bidragit till värdefulla kommentarer om rapportens innehåll och struktur.

Det är min förhoppning att denna rapport kommer att verka till en framtida utveckling av både undervisningsmetodik och utbildningsmaterial inom de systemutvecklingkurser som ges på systemvetenskapliga utbildningsprogram idag.

Borås, augusti 1997



# Innehållsförteckning

<b>1</b>	<b>INLEDNING.....</b>	<b>1</b>
1.1	BAKGRUND.....	1
1.1.1	<i>Organisationen.....</i>	<i>1</i>
1.1.2	<i>Tekniken.....</i>	<i>2</i>
1.1.3	<i>Systemutvecklingen.....</i>	<i>3</i>
1.2	PROBLEMFÖRMULERING.....	4
1.3	PROBLEMAVGRÄNSNING.....	5
1.4	SYFTE.....	5
1.5	FÖRVÄNTAT RESULTAT.....	6
1.6	EGNA RELATIONER OCH FÖRUTSÄTTNINGAR.....	6
1.7	DEFINITIONER.....	7
<b>2</b>	<b>TEORETISK BAKGRUND.....</b>	<b>11</b>
2.1	TIDIGARE FORSKNING.....	11
2.1.1	<i>ISAC-gruppen.....</i>	<i>11</i>
2.1.2	<i>MOA-modellen.....</i>	<i>15</i>
2.2	TEORIER AV BETYDELSE.....	16
2.2.1	<i>Tre perspektiv på användare.....</i>	<i>17</i>
2.2.2	<i>Begränsningar i utvecklingshastighet.....</i>	<i>20</i>
2.2.3	<i>Grupparbete.....</i>	<i>23</i>
2.3	PERSPEKTIV OCH TEORIVAL.....	24
<b>3</b>	<b>METODVAL OCH METODBESKRIVNING.....</b>	<b>27</b>
<b>4</b>	<b>GENOMFÖRANDE.....</b>	<b>29</b>
4.1	PLANERING.....	29
4.2	PRAKTISKT ARBETE.....	29
4.2.1	<i>Litteraturstudie.....</i>	<i>29</i>
4.2.2	<i>Metodtillgänglighet och diskussion.....</i>	<i>29</i>
4.3	PROBLEM.....	30

<b>5</b>	<b>RESULTAT .....</b>	<b>33</b>
5.1	INSAMLAT MATERIAL.....	33
5.1.1	<i>Förändringsarbete och synen på kvalitet .....</i>	<i>33</i>
5.1.2	<i>Användardeltagande i litteraturen .....</i>	<i>34</i>
5.1.3	<i>Systemutveckling ur olika perspektiv.....</i>	<i>35</i>
5.2	RESULTATANALYS.....	38
5.2.1	<i>Förändringsarbete och synen på kvalitet .....</i>	<i>39</i>
5.2.2	<i>Användardeltagande i litteraturen .....</i>	<i>39</i>
5.2.3	<i>Systemutveckling ur olika perspektiv.....</i>	<i>40</i>
<b>6</b>	<b>SLUTSATSER.....</b>	<b>43</b>
6.1	GENERELL METODUTVÄRDERING.....	45
6.2	SAMMANFATTANDE UTVÄRDERING AV ARBETET.....	45
6.3	EGNA REFLEXIONER.....	46
6.4	IDÉER FÖR FORTSATT FORSKNING .....	47

# Figurförteckning

FIGUR 1-1 SAMBANDET MELLAN FÖRÄNDRINGSARBETE, FÖRÄNDRINGSANALYS OCH UTVECKLINGSARBETE .....	8
FIGUR 1-2 FÖRÄNDRINGSANALYS SOM REPETITIV PROCESS. ....	9
FIGUR 2-1 ORGANISATIONEN, INFORMATIONSSYSTEMET OCH INFORMATIONSSYSTEMETS KOMPONENTER.....	12
FIGUR 2-2 EXEMPEL PÅ I-GRAF ENLIGT LUNDEBERG/ANDERSENS INFORMATIONSANALYS .....	13
FIGUR 2-3 MOA-MODELLEN .....	15
FIGUR 2-4 LEAVITTS SAMBANDSMODELL.....	19
FIGUR 2-5 BOEHMS SPIRALMODELL.....	22



# 1 Inledning

## 1.1 Bakgrund

De senaste årens accelererande teknikutveckling har lett till en ökad datorisering på arbetsplatser runt om i landet. På 1960-talet då tilltron till tekniken var som störst, rådde en ganska utbredd uppfattning om att datorisering kunde lösa de flesta problem. Detta ledde ofta till att övergripande organisationsförändringar var nödvändiga för att verksamheten bättre skulle passa tekniken. För den enskilde individen gällde det att snabbt anpassa sin arbetsituation till de nya systemen eller byta arbetsplats. På senare tid har det dock ifrågasatts varför organisationen och dess anställda skall anpassa sig efter datasystemets struktur, och inte tvärtom.

För att lättare förstå den utveckling som skett inom området informationssystem, vilket lett fram dagens situation, följer här en kort historisk bakgrund av tre viktiga faktorer som påverkar införandet av datasystem; organisationen, tekniken och systemutvecklingen.

### 1.1.1 Organisationen

Alltsedan industrialiseringen startade har det alltid varit i arbetsgivarens intresse att de anställda arbetar så effektivt som möjligt. Organisationer växte snabbt i storlek, och det blev allt svårare att detaljstyra den mängd arbetsprocesser som förekom inom verksamheten. Taylor [26] var en av dem som fastställde idéer om hur vetenskaplig arbetsledning skulle ske, genom att effektivisera de verkstadstekniska arbetsprocesser som förekom. Effektiviseringen skedde genom att man fördelade varje process i ett flertal delmoment, som granskades med hänsyn till producerat material på utsatt tid. Genom att kontrollen av arbetet kunde förläggas på datasystem i och med en

ökad teknikspridning, kunde detta även leda till en ökad förstärkning av den byråkratiska strukturen i organisationen.

Denna typ av konventionella produktionssystem bygger på att varje befattning har ett givet arbetsinnehåll. Då de tekniska hjälpmedlen blev alltmer avancerade blev personalens fysiska arbetsinsatser allt mindre. Enligt Edgren et al [2] ledde detta i sin tur till att produktionsgrupper kunde bildas och ansvaret förflyttas till såväl kvalitativa som kvantitativa och ekonomiska mål i produktionen. Detta innebar att den enskilde individens befogenheter ökade, och kompetensutveckling på arbetsplatsen blev en nödvändighet. Man pratar idag ej längre om arbetare, utan nyttjar beteckningen medarbetare i stället. Bland andra Institutet för kvalitetsutveckling [16] har gett ut tretton riktlinjer för hur moderna verksamheter bör utvärderas. En viktig punkt i deras skrift "Utmärkelsen svensk kvalitet" är att varje medarbetare bör ha klara mål, se sin roll i helheten, ha de medel som krävs och kunskap om de resultat som uppnås. Idag används dessa riktlinjer på ett flertal arbetsplatser, bland annat av landstinget [17] som givit ut sitt ledningsinstrument QUL. Denna modell är så gott som identisk i sin struktur till Utmärkelsen svensk kvalitet, och bygger på samma riktlinjer och värderingar (ibid. s 6).

Att utveckla datasystem för den traditionella, byråkratiska organisationen underlättades i och med att varje anställd hade begränsade arbetsuppgifter. Eftersom han/hon utförde dessa uppgifter regelbundet kunde personen i fråga ganska snabbt beskriva sin situation för systemutvecklande personal. Då systemutvecklare ej var intresserade av att bryta den byråkratiska strukturen i organisationen byggdes verksamheten kring det implementerade datasystemet, och användarna som nyttjade det sågs som externa informationskomponenter. Det var med mycket få undantag ej intressant att återge informationsflöden som ej direkt var kopplade till systemet, eller att granska sociala aspekter och följden av ett implementerat system. Fortfarande idag är det enligt Grundén [12] den teknikorienterade systemutvecklingen som är den mest spridda, även om det verkar finnas ett utbrett intresse av en mer humanistisk ansats.

Tillvägagångssättet för att systemutveckla mer decentraliserade organisationer, med individuellt ansvarstagande och arbete i projektgrupper kräver en lite annorlunda ansats. Plötsligt blir det svårt att överblicka varje individs arbetsuppgifter, och möjligheten att se människan som en mindre delkomponent i ett större informationssystem försvåras avsevärt.

### **1.1.2 Tekniken**

Som tidigare nämnts hade tekniken under 1960-talet ännu ej påvisat några större nackdelar. Det upplevdes som önskvärt att införa en vetenskaplig pla-

nering och styrning av produktionen, tillsammans med utvecklingen av totala styrsystem. Målet var att realisera dessa planer med den nya datortekniken som snabbt blev allt mer tillgänglig. Nyttjandet av datorer, som initialt infördes för att automatisera repetitiva manuella rutiner kunde nu verka för att öka ett företags lönsamhet även i de administrativa leden.

En förespråkare för ökad administrativ styrning av personalen var Langefors [20], som menade att informationssystem främst bör nyttjas som kontrollsystem för ett redan operativt system. Denna ökade styrning kunde i sin tur leda till en optimering av verksamhetens lönsamhet. Ett exempel på detta är lagerstyrning, där bundet kapital enkelt kan redovisas över en begränsad tidsrymd.

Problem som snart kom att visa sig berodde på att användarna av datasystemet fick allt svårare att nyttja och lära sig det effektivt då komplexiteten ökade. Avsaknaden av tekniskt kunnande hos personalen innebar samtidigt svårigheter för användarna att uttrycka sina underförstådda behov. En vanlig utbredd uppfattning var enligt Goldkuhl [8] ”Användarna förstår ändå inte. De vet inte vilka behov de har” (ibid. s 11). Det kunde även uppstå situationer där användare kände sig alltför kontrollerade av datasystemen, och helt sonika vägrade att nyttja dem. En teknik för att snabbt engagera användarna i systemet samt få dem att känna sig delaktiga fanns nödvändigt.

Ett komplext datasystem är svårt att hålla enkelt utåt, samtidigt som det bör vara tillräckligt intuitivt och specialdesignat för organisationens syfte. Man kom då på idén att införa användardeltagande genom så kallad *prototyping*. Prototyping innebär att ett schematiskt program skapas över systemet, som redovisar endast den del som användaren i huvudsak kommer att arbeta med. Det är inget krav att ett prototyping program skall ha någon underliggande funktion alls, utan prototypens komplexitet situationsanpassas. Effekter som prototypingen förde med sig var att användares acceptans för systemen ökade då de fått delta vid utvecklingsarbetet. Denna uppfattning anses generellt som sann, och prototyping används fortfarande på bred front. Prototyping som verktyg är dock ej är kopplat till någon specifik metod, och främjar inte i något led designen av organisationsanpassningen i datasystemen.

### 1.1.3 Systemutvecklingen

De första konstruktörerna av datasystem var i huvudsak tekniker och ingenjörer, som utvecklade system som i huvudsak var autonoma. De krävde ingen organisation för att fungera, och människor fungerade som in- respektive utmatningsstationer för hålkort. I Sverige var Langefors [19] en av de första

som kopplade samman organisationslära med datasystemutveckling, och skapade något han kallade för informationssystemteori. Langefors hade idéer om att utveckla systemutvecklingsarbetet till att bygga på en mer teoretisk och vetenskaplig ansats. Han beskriver informationssystemet som ett instrument för att styra det operativa systemet i ett företag, som i sin tur löpande returnerar information. Langefors idéer har använts mycket, bland annat inom undervisning av blivande systemutvecklare i Sverige. Detta syns också tydligt i de systemutvecklingsmetoder som nyttjas inom den akademiska världen, där stora delar av Langefors informationssystemteori fortfarande lever kvar.

Utbildningstekniskt har det även uppstått problem i det att sociala konsekvenser eller psykosociala aspekter vid införandet av informationssystem ej har granskats i någon större omfattning. Då en utbildning för utveckling av datasystem placerats vid en institution som informatik eller informationsbehandling, är det fortfarande idag ovanligt med beteendevetenskapliga kurser inom dessa områden. Uppbyggnaden av tvärvetenskapliga utbildningsprogram som sträcker sig över institutionsgränserna i större omfattning är omständligt att administrera och strukturera. Det kan även vara svårt att motivera en sådan strukturförändring, då de beprövade tekniker som redan finns i drift bevisligen har medverkat till att bygga ett flertal större, driftsmässiga datasystem. Då metoderna fungerar, varför ändra på dem?

## 1.2 Problemformulering

Då en organisation bedriver ett kontinuerligt och kvalitativt förändringsarbete är det en självklarhet att samtliga anställda som vill delta skall ha rätt till detta. Arbetet brukar ofta delas in i projektgrupper, där arbetet kan effektiviseras genom att samtliga anställda får uttrycka sig i egna ord om sin arbetssituation. I de fall då detta leder till en förändring eller konstruktion av ett datasystem borde det tyckas självklart att samtliga medarbetare bidrar med samma konstruktiva information till designen av detta. Allt för att en integration med befintlig arbetssituation skall gå så smidigt som möjligt.

Detta arbete kan dock förhindras i och med att extern systemutvecklande personal inkallas som konstruerar datasystem baserat på principen av det totala styrsystemet som diskuterats innan. Det är vanligt att medarbetare i organisationen klassificeras, och att endast vissa ”intressanta” individer väljs ut för systemutvecklingsarbetet. Eftersom användare enligt denna uppfattning endast ses som mottagare och sändare av information är det oftast endast intressant att se i vilka kretsar användaren arbetar, samt vilka informa-

tionsmängder han/hon är involverad i. I vissa sällsynta fall tillfrågas inga användare alls under systemutvecklingsprocessen.

Då det förekommer mycket diskussion i huruvida detta synsätt på användare är att föredra vid systemutveckling, blir det intressant att undersöka hur större systemutvecklande företag bygger sina system. Sammanfattat resulterar detta i följande fyra huvudfrågor:

- Bör användare delta vid systemutvecklingsarbete, och i så fall varför?
- Vilka fördelar respektive nackdelar finns det i att låta användare delta i arbetet?
- Inom vilka faser i utvecklingsarbetet bör man låta användare delta?
- Hur organiseras processarbetet?

### **1.3 Problemavgränsning**

Det faktum att det finns ett oräkneligt antal systemutvecklingsmetoder endast i Sverige och att dessa konstant vidareutvecklas gör att det är en omöjlig uppgift att genomföra en total sammanställning av varje metod och dess tillvägagångssätt. Jag har valt att fokusera på tre större svenska systemutvecklingsmetoder, varav en används flitigt inom den akademiska världen och de övriga två av Sveriges större systemutvecklande företag. De tre metoderna är samtliga verksamhetsorienterade och går under namnen VIBA, DELTA och EFSOS. En litteraturstudie kommer även göras beträffande ett flertal fallstudier vid införandet av datasystem och dess konsekvenser.

På grund av upphovsrättigheter så kommer ingen detaljinformation om respektive metod att redovisas. Tillräckligt med information kommer att återges för att korrekt återge metodens synsätt och tillvägagångssätt, dock så bör det ej vara möjligt att återskapa hela eller delar av metodens faser efter att ha studerat rapporten.

### **1.4 Syfte**

Studien genomförs i syfte att klarlägga användarens roll vid systemutvecklingsarbete i Sverige. Granskningen görs avseende deltagandets syfte, tillämpning och tillfälle.

Målgruppen för studien är alla som idag är involverade i förändringsarbete, oavsett om det rör sig om införandet av datasystem eller förändring av organisationsstruktur. Då rapporten redovisar tillvägagångssättet i ett flertal me-

toder finns det även en potentiell målgrupp i lärare och studenter som endast haft tillgång till akademiska systemutvecklingsmetoder och vill bredda sitt kunskapsområde.

## 1.5 Förväntat resultat

Det förväntade resultatet är en översikt över riktlinjer och tillämpningar av användarmedverkan vid systemutveckling i Sverige. Motsvarar synen på människan som kompetent, flexibel och självständig medarbetare inom moderna, kvalitetsstyrda organisationer den uppfattning som idag existerar inom systemutvecklande företag? Vilka fördelar kan skönjas i det implementerade systemet då användarmedverkan varit en påverkande faktor för systemets struktur och design?

Även tillvägagångssättet vid användarmedverkan i respektive metod kommer att redovisas. Allt för att bättre förstå de olika metodernas syfte med att ta in användare i arbetet, och för att bättre ge en bild av begränsningar och fördelar med respektive förfarande.

## 1.6 Egna relationer och förutsättningar

Då jag fram till mina högskolestudiers start arbetat som applikationsutvecklare i olika sammanhang, har speciellt frågor gällande användbarhet i implementerade system intresserat mig. Huvudintresset var att utveckla data-system som var så pass användbara och funktionella att deras existens aldrig skulle ifrågasättas. Ett av de system jag utvecklat var ett mindre ekonomisystem, som senare övergavs för ett standardpaket. De främsta orsakerna var att mitt system var modellerat efter en bild av verksamheten som organisationen snabbt växte ur. Systemet blev för litet och klarade ej de tyngre belastningarna. Då systemet var utvecklat i ett generellt utvecklingsverktyg som ej var objektorienterat, innebar det att tillägg av nya komponenter tillförde en ökning av komplexiteten i systemet som var mycket större än linjär<sup>1</sup>.

Efter en tids studier föll jag bort från tankarna vad gällde nyttjandet av data-system, och blev liksom övriga studenter ganska besatt i tanken att få klart projektutvecklade system på så kort tid som möjligt. De simulerade användarna av systemet tog ingen någon större hänsyn till, och all projektutveckling byggde på principen att skapa en statisk avbild av en verksamhet för

---

<sup>1</sup> se Brooks [1] för en diskussion om detta fenomen

efterföljande implementation. Någon diskussion vad gällde föränderliga organisationer eller användarmedverkan vid systemutveckling förekom inte alls. Det var först när jag efter mitt andra år började arbeta med att utveckla ett mindre intranetsystem på skolan som intresset återigen väcktes för användardeltagande i systemutvecklingen.

En projektgrupp kallad ITCentrum bildades, och målet med verksamheten var att bedriva studier på hur implementerade datasystem påverkade den organisationella strukturen och kommunikationen inom större organisationer. Problemet var bara vilket angreppssätt som skulle nyttjas för att få in systemen i verksamheten, och utöver detta få folk att nyttja dem dagligen. Två av de mest nyttjade systemen som jag personligen implementerat är högskolans databasstyrda nyhetssystem tillsammans med ett system för att administrera och redovisa kursplaner. Dessa två projekt var från början idéer som anställda på högskolan haft, och under utvecklingens gång testades systemens funktionalitet under en större tidsperiod för att vara säker på att man uppfyllt de förhoppningar som från början skapade förslagen.

Under arbetet med att implementera systemen dök det upp en rad frågor. Då datasystemet till högskolans hemsida utvecklades skapade jag även grafikbilderna på framsidan tillsammans med designkonceptet, i samråd med en tillkallad projektgrupp. Denna projektgrupp arbetade sedan fram generella strukturer för designen. De anställda på högskolan involverades inte förrän arbetet var färdigt, trots att det var deras arbetsplats profil utåt som fastställdes. Frågan var här hur man skulle gå till väga för att involvera fler personer på ett tidigt stadium. Under utveckling av kursplanssystemet involverades ingen anställd under utvecklingens gång, utan hela projektet bedrevs med initiala önskemål i bakhuvudet. Allt för att få färdigt produkten på så kort tid som möjligt. Det är idag omöjligt att säga hur produkten skulle sett ut om den utvecklats under en något större tidsrymd i samråd med t.ex. samtliga institutionssekreterare.

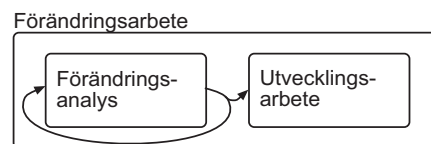
Då jag efter genomfört kursprogram på kandidatnivå fortfarande kände mig osäker hur slutanvändare bör bemötas vid systemutveckling, bestämde jag mig för att göra en praktisk studie av systemutvecklande företags tillvägagångssätt i denna rapport.

## **1.7 Definitioner**

När man diskuterar termer inom systemutveckling så finns det egentligen endast en sak de flesta är överens om, och det är att det existerar mycket mångtydighet inom begreppen. Begreppet "system" i samband med infor-

mationssystemutveckling är ganska opreciserat, och när man talar om utveckling av datasystem så finns till exempel inget som förutsätter att systemet skall interagera med personer överhuvudtaget<sup>2</sup>. Detta gör att det blir svårt att sätta likhetstecken mellan datasystemutveckling och generellt systemutvecklingsarbete.

Sambandet mellan förändringsarbete, förändringsanalys och systemutvecklingsarbete åskådliggörs utöver definition även i figur 1-1. Bilden är baserad på ett strukturschema återgivet av Goldkuhl/Röstlinger [9]. Notera att det som i bilden redovisas som utvecklingsarbete kan innehålla systemutvecklingsarbete, men behöver ej göra det.



**Figur 1-1 Sambandet mellan förändringsarbete, förändringsanalys och utvecklingsarbete**

Övriga termer som kan uppfattas som tvetydiga tar jag härmed upp enligt nedan. Min definition av termerna är redovisad mest för att fastställa mitt förhållande till respektive begrepp, och understöds inte av någon speciell teori om så ej är angivet.

**ADB-System.** För att beteckna ett datasystem som utvecklas verksamhetsorienterat används beteckningen ADB-System. Dessa system brukar även kallas för databaserade informationssystem.

**Användare.** Användare är de personer som slutligen kommer att nyttja det driftsmässiga datasystemet. Detta kan röra sig om den mer traditionella synen på arbetare eller den mer moderna medarbetaren.

**Användarmedverkan.** Med användarmedverkan menas möjligheten för en organisations medarbetare att delta i förändringsarbete. Detta innebär ej endast fysisk närvaro, utan synen på medarbetare baseras på en person som är dynamisk och kreativ med ett brett kunskapsområde vars kunnande är värdefullt i alla utvecklingsfaser. Användarmedverkan innebär möjlighet att påverka resultatet av en pågående förändringsprocess.

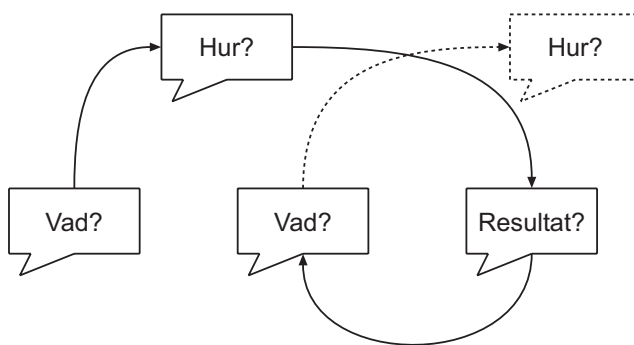
---

<sup>2</sup> Exempel på detta är program som under regelbundna tidsintervall organiserar data

**Användarcentrerad systemutveckling.** Det finns ett flertal definitioner i vad det innebär att arbeta användarcentrerat. Enator skriver i sin metod PAS [4] att användarcentrerad systemutveckling riktar sig framförallt till den person som kommer att nyttja systemet för att lösa sina arbetsuppgifter. Detta gör att mycket tid läggs vid användbarhetsstudier och utbildning. Den definition jag använder för användarcentrerad utveckling är dock att fokusering på användaren i centrum bör kunna nyttjas i alla led inom systemutvecklingsprocessen. Då en fas i arbetet kräver att informationsstrukturen ses ur informationssystemets synvinkel så har man avvikit från det användarcentrerade synsättet. Detta eftersom det enligt min uppfattning är svårt att se till individuella behov vid en översiktsvy av det totala informationssystemet.

**Datasystem.** Ordet system betyder sammanställning, och består generellt av ett antal komponenter som samverkar i en större helhet. Beteckningen datasystem används för att åskådliggöra ett system där en eller flera av komponenterna är datorbaserade. Datasystem behöver ej vara verksamhetsorienterade.

**Förändringsanalys.** Enligt Goldkuhl/Röstlinger [9] så innebär förändringsanalys att man utvecklar beslut om det fortsatta förändringsarbetets inriktning, där man ej bör ta några förändringsåtgärder för givna. Denna beskrivning gäller för de allra flesta angreppssätt, det som främst skiljer olika metoders tillvägagångssätt är sättet åtgärderna diagnostiseras. Då somliga förespråkar en problemsökande ansats finns det ett flertal som uppmuntrar en mer positivistisk syn. Se figur 1-2 för en skiss av hur Institutet för kvalitetsutveckling [16] åskådliggör den ständigt upprepande processen förändringsanalys.



**Figur 1-2 Förändringsanalys som repetitiv process enligt Insitutet för kvalitetsutveckling [16].**

**Förändringsarbete.** I moderna organisationer sker ett kontinuerligt, ständigt utvecklande förändringsarbete. Förändringsarbete är det verksamhetsorienterade arbete som utförs i syfte att förbättra organisationsmässig kvalitet, förbättra arbetsmiljön för de anställda samt att hjälpa organisationen nå sina verksamhetsmål vid minimerad kostnad. Detta involverar en upprepad analysfas, som kan leda till förändringar i organisationsstrukturen eller förändringar och införandet av datasystem.

**Informationssystem.** Det nätverk av informationsentiteter inom en organisation som generellt används för att skapa en schematisk avbild av verksamheten. Informationssystem används för insamling, bearbetning, lagring och distribution av information. Ett informationssystem kan ej förutsätta ett datasystem, samtidigt som förekomsten av datasystem inom en organisation förutsätter att det existerar ett informationssystem.

**Medarbetare.** Medarbetare är dagens och morgondagens dynamiska individer som bygger upp organisationen. Dessa personer bör ha kunskap inom ett brett kunskapsområde, där arbetsuppgifter kan komma att inkludera allt från val av råmaterial och produktion till test, marknadsföring och försäljning. De bör även ha grundläggande finansiella kunskaper, tillsammans med kunskaper om att sätta mål, mäta kvalitet på organisationens produkter eller tjänster samt finna svagheter och bestämma åtgärder för att motverka dessa. Det är också viktigt att ha kunskap om hur den totala verksamheten är strukturerad. Medarbetare är en stark kontrast till den mer traditionella byråkratiska organisationens arbetare, som endast hade kunskap inom ett begränsat område och som ej såg mer av organisationen än sin egen avdelning.

**Objektorienterad.** Abstraktion av information och processer till inkapslade komponenter och kommunikationen dem emellan kallas vanligtvis för objektorientering.

**Online.** För att beteckna ett system som kan brukas av ett flertal personer samtidigt, lokalt eller på distans via webbläsare nyttjas beteckningen onlineprodukt.

**Prototyping.** Ordet prototyp betyder första exemplar av något, och prototyping är den process då detta första exemplar testas avseende funktionalitet och korrekthet. I datasammanhang brukar en prototyp motsvaras av ett dataprogram.

**Systemutveckling.** Med systemutveckling brukar man generellt sett mena datasystemutveckling, det vill säga skapandet av informationssystem med en eller flera datorkomponenter involverade.

**Utvecklingsarbete.** Det arbete som verkar för att realisera förändringar inom en organisation. Detta arbete kan innehålla systemutvecklingsarbete, men behöver ej göra det. Se för övrigt figur 1-1.

## 2 Teoretisk bakgrund

### 2.1 Tidigare forskning

Sverige har en ganska bred profil av systemutvecklare som arbetar verksamhetsorienterat. Studier och forskning utförda av dessa gällande användarcentrerad användarcentrerad har bedrivits ända sedan 1960-talet, och pågår än idag. För att sammanfatta de generella värderingsskiften som skett under åren redovisas nedan två resultat från olika forskningsgrupper. Den första är ISAC-gruppen, som i Langefors andas<sup>3</sup> utvecklade ett flertal metoder baserade på användarorienterade systemutvecklingsmodeller. ISAC-gruppens metoder används fortfarande inom ett flertal institutioner i Sveriges högskolor, och har haft stort inflytande i utvecklingen av både akademiska systemutvecklingsmetoder såsom företagsanpassade specialmetoder. MOA-modellen som presenteras senare är skapad av fil. dr. Kerstin Grundén, och bygger på mer än tjugo års forskning och praktiskt arbete. Syftet med modellen är att utveckla ett mer människoorienterat perspektiv på systemutveckling, som till mycket baseras på människans möjlighet till utveckling i arbetet.

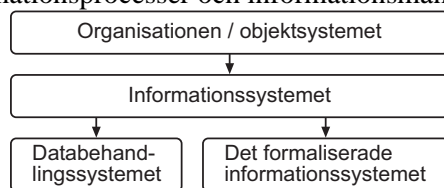
#### 2.1.1 ISAC-gruppen

Då Langefors idéer fått fäste i den akademiska världen fanns det ett flertal systemutvecklare som såg ett behov av att utveckla dessa vidare för att ge

---

<sup>3</sup> Se kapitel 1.1.3

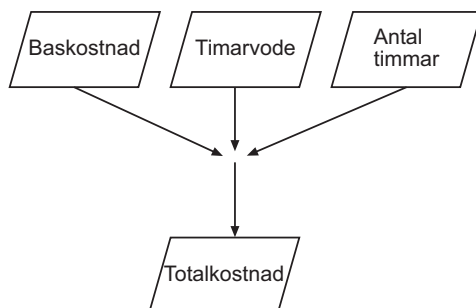
dem en djupare innebörd. Detta ledde till att en forskningsgrupp bildades under tidigt 1970 tal som gick under namnet ISAC-gruppen. Målet med ISAC-gruppens arbete var att göra Langefors idéer praktiskt användbara i en metod, samtidigt som en utvidgning av begreppen skulle ske. 1975 släppte två medlemmar i gruppen, Lundeberg/Andersen [22] boken ”Systemering – informationsanalys” som framför allt fokuserade på informationssamband mellan olika informationsprocesser och informationsmängder.



**Figur 2-1 Organisationen, informationssystemet och informationssystemets komponenter**

Lundeberg/Andersen redovisade i boken sin bild av organisationen där centrumunkten var det totala informationssystemet (se figur 2-1). Informationssystemets uppgift var att förse verksamheten med information, och definieras av Lundeberg/Andersen [22] som ett system för insamling, bearbetning, lagring och distribution av information. Denna beteckning av informationssystem anses av många fortfarande vara den generellt mest spridda. Informationssystemet bestod i sin tur av ett formaliserat informationssystem som konceptualiserade sättet informationssystemet bearbetade sin information. De uppgifter som kunde utföras utan mänsklig kännedom (t.ex. registrering av stokastiska data och tidsstyrd sortering av poster) utfördes i sin tur av det automatiserade databehandlingssystemet.

Lundeberg och Andersen var även bland de första som satte en definition av vad systemering respektive systemutvecklare innebar. Systemering var det arbete som fokuserade på analys och design av informationssystem. Systemerare eller systemutvecklare ansågs då vara den yrkeskategori som bedrev detta arbete. För att bedriva detta arbete utvecklades en metod som gick under namnet informationsanalys (se figur 2-2).



### **Figur 2-2 Exempel på i-graf enligt Lundeberg/Andersens informationsanalys**

Informationsanalysen bygger på att grafiskt åskådliggöra de informationsmängder som förekommer i systemet, tillsammans med de processer som krävs för att relatera dem emellan. I figur 2-2 består processen underförstått av knutpunkten mellan in- respektive utgående informationsmängder. I-grafen redovisad enligt figur 2-2 är den mest abstrakta formen av grafer, och kan brytas ner i mindre delgrafer efter behov. Även här syns tydligt influenserna från Langefors som ansåg att det endast var möjligt att överblicka en verksamhet genom att dela in den i ett flertal mindre delsystem.

Då verksamheten är redovisad i ett flertal i-grafer så tjänstgjorde sedan informationsanalysen som grund för programutvecklingen. Det är viktigt att notera att användarna av systemet ej förekommer inom i-graferna, utan riskerar att förbli osynliga fram tills den dagen då systemet tas i drift och eventuella komplikationer kan uppstå. Detaljer som fördelning av arbetsbörda människa/maskin eller sociala konsekvenser vid implementation av ett datasystem riskerar helt att förbises.

Grundén [11] skriver att de flesta metoder som utvecklats inom systemutvecklingsområdet i huvudsak fokuserar på analys av information och abstrakta rutiner i verksamheter, och att dessa i sin tur generellt osynliggör användarna som subjekt. ISAC-gruppen var medvetna om sina brister i informationsanalysen och presenterade 1978 i boken ”Systemering” en variant av den ursprungliga ISAC modellen i en mer användarorienterad inriktning. De personer som sammanställde boken denna gång var Lundeberg, Goldkuhl och Nilsson [21].

Från att ha varit en systemteoretiskt inriktad systemutvecklingsmetod<sup>4</sup> sågs nu informationssystemet som en förmedlare av information mellan människor. En ny fas i systemeringen har också lagts till, förändringsanalysen. Dess uppgift var att granska om ett databaserat informationssystem verkligen var lösningen på de problem som kunde föreligga inom organisationen. Hela ansatsen i förändringsanalysen baseras på att initialt ta de problem som existerar och vända de till något positivt. Även om målet med den nya ISAC modellen var en mer humaninfologisk ansats så kretsade fortfarande utvecklingsarbetet kring informationssystemet där människorna sågs som mottagare- och sändarkomponenter av information. Göran Goldkuhl [8] som deltog i ISAC-gruppens arbete under ett flertal år beskriver än idag användare av datasystem enligt:

---

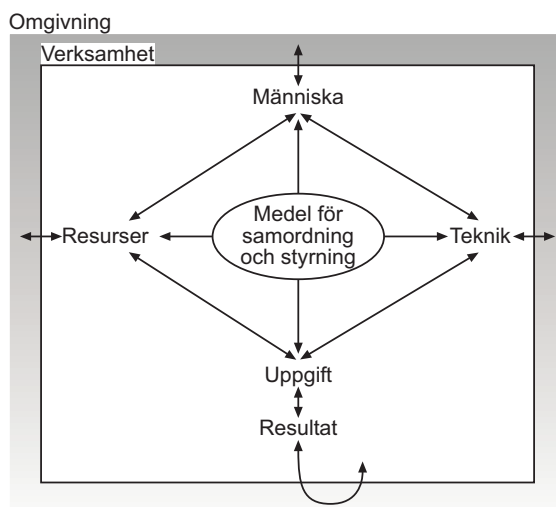
<sup>4</sup> Se kapitel 2.2.1 för en detaljerad beskrivning av de perspektiv som föreligger

”Inanvändare är sådana personer som lämnar ifrån sig (in)meddelanden till datasystemet. Utanvändare är sådana personer som mottager (ut)meddelanden från ett datasystem och utför handlingar på basis av dessa meddelanden” (ibid. s 16)

### 2.1.2 MOA-modellen

En av de personer som ifrågasatt lämpligheten i att låta verksamheten och användarnas arbetssituation kretsa kring ett centraliserat informationssystem är Kerstin Grundén. Grundén [11] har arbetat aktivt som systemutvecklare och studerat framförallt de sociala aspekterna vid införandet av databaserade informationssystem. En av Grundéns huvudteser är att det råder brist på teori- och metodansatser som belyser samspelet mellan människa, organisation och ADB-system, och har därför utvecklat en egen ansats som understödjer detta synsätt.

MOA-modellen består av fyra grundläggande komponenter; människa, teknik, uppgift och resurser. Dessa kommunicerar fritt genom olika medel för samordning och styrning. Det unika med modellen är att en verksamhets centrala punkt anses vara på det sätt som samordningen och styrningen sker, genom en väl definierad organisationsstruktur eller ADB-systemutformning. Människorna i organisationen betraktas ur en processororienterad synvinkel, i det att *människorna* nyttjar *resurser* och *teknik* för att utföra sina *arbetsuppgifter* i syfte att producera *resultat*. MOA-modellen är illustrerad i figur 2.3.



Figur 2-3 MOA-modellen (Efter *ibid.*, s 129)

Modellens medel för samordning och styrning innehåller ett flertal komponenter. Utveckling av databaserade informationssystem kan betraktas som en sådan eftersom denna typ av system är en del både av verksamhetens struktur, kultur och processer. Systemen kommunicerar även med samtliga delar i verksamheten vilket leder till en påverkan av samtliga i verksamheten förekommande delmängder.

Då det är människor som utformar den centrala styrningen i organisationen finns det enligt Grundén olika synpunkter om lämpligaste sätt att utföra detta på. Då de flesta organisationer av tradition är av hierarkisk struktur innebär detta att det förekommer medarbetare med olika ansvarstagande och beslutbefogenheter. Även om samtliga röster skall väga lika tungt utspelas det enligt Grundén ofta en maktkamp om resursfördelningen. I sig kan organisationens utformning ses som en politisk process där olika intressen, meningar och maktförhållanden konfronteras.

Grundén har även utgått från sin modell och skapat ett perspektiv, kallat MOA-perspektivet. Detta människoorienterade perspektiv betraktar verksamheten ur människans synvinkel och lyfter fram sambanden mellan medarbetarens möjligheter till utveckling i arbetet, utformningen av administrativa ADB-datasystem och organisationen. En av de viktigaste aspekterna vid utformningen av ett människoorienterat system är enligt Grundén att människorna inom organisationen fungerar väl tillsammans med användningen av ADB-systemen.

Grundén anser i den sammanfattning av MOA-modellen och perspektivet som utgivits i samband med arbetsmiljöfonden och NUTEK [10] att synsättets syfte främst borde kunna nyttjas som en vägvisare för en utveckling till en förnyad, effektivare organisation samt utveckling av människan i arbetet:

”Det människoorienterade synsättet kan användas som en allmän kunskapsbas och inspirationskälla i systemutvecklingsarbetet. Troligtvis får det en reell betydelse för systemutvecklingen först när kunskaperna omsätts i praktiskt tillämpbara metoder” (ibid. s 35)

Grundén anser att det kvarstår ganska mycket forsknings- och utvecklingsarbete innan en metod kan formars som till fullo understödjer det mer människoorienterade synsättet.

## **2.2 Teorier av betydelse**

Då forskning inom systemutvecklingsmetoder med åren verkar ta mer hänsyn till de sociala aspekter som föreligger, kan man fråga sig varför denna aspekt på utvecklingsarbetet ej tagits tidigare. Frågan väcks också varför inte samtliga systemutvecklare idag tillämpar humanistiska ansatser om dessa verkar vara de som bäst bemöter människorna i organisationen.

För att sammanfatta de förändringar i synsätt på organisationen och människan som skett över åren kommer här att redogöras för de tre perspektivskiften som Markku Nurminen [24] identifierat. Dessa synsätt har starkt influerat designen av de metoder som nyttjas i de flesta organisationer. Utöver detta kommer även tänkbara orsaker till utvecklingstekniska problem redovisas, som kanske inte borde vara en faktor som påverkar tillvägagångssätt vid systemutvecklingsarbete med finns där likväl.

Slutligen så kommer en kort sammanfattning över grupparbetets vara eller inte vara att redovisas. De faktorer som nämns här är viktiga vad gäller förståelsen av interaktionen mellan beställare och leverantör av informationssystem, då utvecklingsarbetet ofta sker i isolerade grupper.

### **2.2.1 Tre perspektiv på användare**

En person som arbetat mycket med att beskriva förhållandet mellan ny teknik och människorna som nyttjar den är Markku Nurminen. Nurminens [24] fundamentala fråga som forskningen är baserad på är vad tyngdpunkten ligger mellan människa och maskin. Nurminen har därför identifierat tre perspektiv där tyngdpunkten i det så kallade systemteoretiska perspektivet ligger hos maskinen, i det sociotekniska perspektivet balanseras de båda intressefaktorerna och i det humanistiska perspektivet sätts alltid människan i centrum. Perspektiven betecknar enligt Nurminen den förändring i synsätt som skett bland systemutvecklare, och har utvecklats i kronologisk följd baserat på tidigare händelser. Dessa perspektivs synsätt är det mycket viktigt att vara medveten om, då det påverkat utvecklingen av de systemutvecklingsmetoder som används än idag. Då ISAC metoden formades på 1970-talet rådde en ganska utbredd uppfattning enligt det systemteoretiska perspektivet. Under 1980-talet då Goldkuhls VIBA och första versionen av WM-Datas DELTA handbok sammanställdes rådde ett utbrett sociotekniskt perspektiv, och nu under 1990-talet då Enators EFSOS sammanställdes i sin nuvarande, objektorienterade form råder en ganska utbredd humanistisk syn på människor och kvalitet.

Nurminens perspektiv beskriver synsättet på människa och system enligt tre idealtyper, vilket innebär att endast ett fåtal av de händelser som inträffar i verkligheten tas upp. Dessa händelser utvecklas sedan vidare i detalj. Ett exempel som Nurminen ger på sid 18 är den byråkratiska strukturen beskriven av Max Weber, där den funktion som ansetts mest fundamental för en formell organisation är dess regler. Dessa regler struktureras sedan till deras mest abstrakta form, där organisationens funktion granskas som om den verkligen existerade, eller kunde existera i en sådan form. Perspektiven som

anses följa varandra i kronologisk följd kommer kort att redovisas i korrekt ordning nedan.

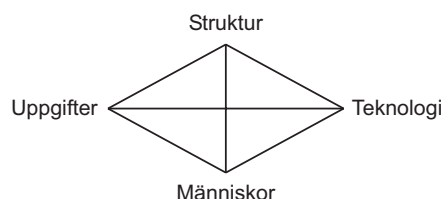
Det *systemteoretiska perspektivet* bygger på en bild av verksamheten där idealtypen är ett integrerat, totalt system. Styrelsen/ledningen inom organisationen nyttjade informationssystemet som medel för styrning av de anställda. Människorna utför sina arbetsuppgifter optimalt och efter anvisningar från datasystemet. Då människors kunskap anses kunna lagras i systemet i form av information innebär detta att kunskap enligt perspektivet ej innehåller värderingar utan endast fakta. Då informationen anses vara objektiv kan den tas ur sitt samband och alltid lagras centralt inom datasystemet.

Människan har två roller i det systemteoretiska perspektivet beroende på modell. I det första fallet så exkluderar informationssystemet användaren och människan betraktas som en extern hämtare av data. Användaren kan aldrig påverka funktionaliteten i systemet och blir således helt beroende av dess struktur. I det andra fallet tas människan med som en komponent i systemet, men ses då mest som ett rationellt och passivt kugghjul. Då användaren ansluts till systemet enligt löpande band principen uppstår ett visst beroende mellan användare och datasystem. Människor kommunicerar ej med varandra utan all kommunikation sker endast till och från systemet genom lagring och bearbetning av information.

Då datasystem utvecklas enligt detta perspektiv utgick man från principen att verkligheten alltid gick att återskapa i ett system, det gällde endast att hitta den korrekta modellen. När ett system väl var implementerat och i drift var det färdigt och vidare utveckling skulle ej behöva ske. På den tiden det systemteoretiska perspektivet nyttjades fanns det fortfarande datorer som programmerades genom fysisk kabeldragning, vilket kunde leda till enorma strukturförändringar vid mindre förändringar i den operativa strukturen.

Det *sociotekniska perspektivet* skapades som svar till de problem som uppstod genom det systemteoretiska perspektivet, särskilt genom frågan hur individens arbetssituation ställdes i perspektiv till systemet. Idealtypen som växte fram blev av dual karaktär, där hänsyn togs till både sociala och tekniska problem. Data som lagras inom systemet blir till information när den tolkas av en människa, och människan har sedan möjlighet att tillgodogöra sig denna information som kunskap. All information som lagras inom systemet måste ha en mening, och objektiviteten är ej i sig ett självändamål längre då man ansåg det nödvändigt att få med syfte i informationsdelmängderna. Då information lagrad i systemet skulle presenteras för användare med olika krav och förutsättningar växte även behovet av olika datavyer fram. Organisationsstrukturen i det sociotekniska perspektivet brukar ofta baseras på Leavitts diamantstruktur (Nurminen [24] s 74) där människor,

teknik, uppgifter och organisationsstruktur samverkar, se figur 2-4. Denna struktur är mycket använd, och en modell som baserats utifrån den är MOA-modellen återgiven ovan (se figur 2-3). Det den framförallt vill åskådliggöra är att om en variabel ändras i organisationen så påverkas de övriga tre automatiskt.



**Figur 2-4 Leavitts sambandsmodell (Efter Nurminen [24], s 74)**

Det sociotekniska perspektivets syn på människan grundar sig på bilden av X- och Y-människor, som enligt Nurminen (ibid. s 103) tidigare definierats av McGregor. X-människan är av naturen lat, arbetar så lite som möjligt, har inga ambitioner, ogillar ansvar och föredrar att bli styrd. Y-människan är motsatsen till X-människan, är aktiv och gillar ansvar. Den sociotekniska synen bygger helt på Y-människan, och ser även människors aktiviteter som gruppaktiviteter. Då människan i organisationen behöver information för att skapa meningsfulla arbetsuppgifter samt för att informera övriga medarbetare växte behov på informationstillgänglighet *ad hoc* fram. Icke-officiell information ej nödvändig för verksamhetens struktur granskades också, eftersom denna visat sig viktig för människors trivsel och tillfredsställelse, vilket i sin tur ledde till bättre arbetsprestationer.

För första gången tilläts användare att delta och komma med synpunkter och kommentarer under systemutvecklingsarbetet. Restriktionen var dock att integriteten inom systemet ej fick påverkas som en följd av detta. Användarens roll i detta kunde sträcka sig från att endast få en utbildning i det färdiga systemet till att bestämma över systemets utseendemässiga design. Viktigt är dock att användaracceptans av implementerade system nu värderades högt. Lundeberg et. al inom ISAC-gruppen [21] var några av de som starkt hävdade att det ej räckte med ett tekniskt fulländat system. Om det ej accepterades av sina användare så går det ej heller att realisera de förväntade målen med systemet. En intressant aspekt Nurminen har om användardeltagande är att det är anmärkningsvärt att det förekommer en diskussion angående användardeltagande vid systemutveckling överhuvudtaget, som om aktiviteten i fråga ej var medarbetarnas från början.

Fortfarande bygger systemutvecklingsarbetet på en bild av organisationen som statisk rörelse, då systemen ansågs för stora för att överblicka och förändra. På grund av systemens komplexitet och mån av integritetskontroll

tillåts ej användare att förändra något i det implementerade systemet, utan målet var att producera en bra och användbar slutprodukt med så små förändringar på så lång tid som möjligt.

Som den kanske slutliga protesten mot informationssystemens styrning av människorna utvecklades det *humanistiska perspektivet*. Målet och idealiteten med perspektivet är att samtlig lagring, bearbetning och överföring av data sker individuellt och lokalt hos varje anställd. Detta för att undvika all form av kontroll som kan utövas genom ett centralstyrt system. Kunskapsbegreppet har vidgats ytterligare, och man definierar nu kunskap i tre större termer som påståendekunskap, förtrogenhetskunskap och färdighetskunskap. Av dessa tre kunskapstyper är det möjligt att lagra påståendekunskap i ett datasystem.

I stället för arbetstilldelning låter organisationen användarna själva finna motivation att arbeta. Varje människa är en renodlad Y-typ och tillåts fritt påverka sin arbetssituation då detta i sin tur leder till individuell tillfredsställelse. Ingen kommunikation sker längre med systemet, utan all kommunikation och koordination sker medarbetare emellan. Människorna kan dock nyttja systemet för kommunikation med varandra.

Då varje användare är ansvarig för sin egen arbetsmiljö, är också användaren ansvarig för sitt eget datasystem. Utvecklingen av detta sker gradvis, steg för steg så att de hinner förstå och lära sig från den specialutbildade systemtekniska personalen. Då varje användare ser över och utvecklar sitt eget system försvinner även behovet av en gemensam datacentral. Hela verksamheten byggs upp av individer som arbetar mot gemensamma mål via kommunikation.

### **2.2.2 Begränsningar i utvecklingshastighet**

De perspektiv som hittills nämnts fokuserar i huvudsak på fyra entiteter; organisationen, människan, tekniken och informationssystemet. Problem med användaranpassning av programvarorna anses generellt kunna lösas med större deltagande vid systemutvecklingsprocessen. Det är dock inte någon av metod- och perspektivutvecklarna vilka hittills är nämnda som diskuterar användardeltagandes betydelse för själva produktutvecklingen. Då tekniken och organisationsstrukturen utvecklats i ett mycket snabbt tempo så har de datorbaserade hjälpmedel som används för implementation av datasystemen skett i en mycket långsammare takt. Systemutvecklingsmetoderna skapar strukturer och relationer dem emellan som är tänkta att ligga till grund för vidare produktutveckling, utan egentlig diskussion om verktygen verkligen klarar av att leva upp till systemutvecklarnas förväntningar. Gö-

ranson/Ottersten [13] är några av dem som konstaterar att det finns ett dilemma då avbilden av organisationen skall överföras till ett datasystem:

”När applikationen skall kodas skapas begrepp som ofta har mycket vagt samband med de begrepp som används i verksamheten. Det blir då tämligen svårt att finna sambanden mellan de företeelser som existerar i verkligheten och den data, det beteende och den presentation som företeelsen motsvarar i det färdiga informationssystemet” (Görranson/Ottersten [13] s 20)

Görranson och Otterstens objektorienterade metod för verksamhetsmodellering, COOL-metoden delar därför upp systemutvecklingsarbetet i två huvudsakliga områden; projekttaktik och utvecklingstaktik. Projekttaktiken beskriver sättet projektet bör bedrivas, samtidigt som utvecklingstaktiken syftar till att hjälpa utvecklaren att ”...skapa ett formaliserat resultat, en avbild, utifrån ett oformaliserat underlag, en verklighet” (ibid. s 125). Metoden nämner att det behövs designhjälpmedel vid denna process, men diskuterar inte hur de skulle se ut eller nyttjas.

I de fall då datasystemet utvecklas utifrån en statisk avbild som gjorts av organisationen, och den i sin tur förväntas vara relativt oföränderlig med tiden så anses det ej lika avgörande vilket produktutvecklingsverktyg som används. Ur perspektivets ofta idealtypiska bild så finns det ej heller begränsningar i utvecklingstid eller underhållsproblem. Enligt Grundén [11] så finns mycket få, om några alls, metoder som granskar effekterna efter implementerat system, och detta inkluderar även underhåll och support av datasystemen.

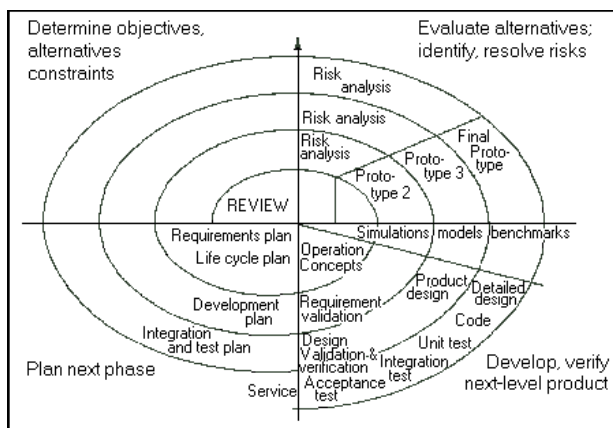
En person som på bred front diskuterat just utvecklingshastighetens dilemma är Frederic Brooks [1], som 1987 publicerade en rapport som väckt mycket stor diskussion, ”No Silver Bullet – Essence and Accident in Software Engineering”. Brooks delar upp utvecklingsproblemen i två områden; väsentliga och oväsentliga fel (essence and accident). Väsentliga fel är övergripande strukturella misstag och brister, där de oväsentliga felen kan vara inmatningsfel eller problem att formulera strukturen som programkod.

Brooks ståndpunkt är att de väsentliga felen vid produktutveckling är de som orsakar mest skada. Detta eftersom det då Brooks skrev sin rapport saknades verktyg som underlättade den logiska designen av datorprogram. Skadan ansågs vara så pass stor att begreppet ”Software Crisis” myntades, som betecknade de allmänna problemen med att producera dataprogram i tillräcklig hastighet för att mätta efterfrågan. Viktiga aspekter att ta hänsyn till vid design av datasystem är enligt Brooks att de skall vara anpassade efter sina användare samt vara lätta att förändra. Brooks skriver också att då arbets-

miljön förändras så utvecklas också användarnas krav på mjukvaran, och de ser nya möjligheter till redan implementerade funktioner.

Brooks har fått kritik för sina åsikter under åren, och det finns ett flertal personer som hävdar att just deras utvecklingsverktyg är lösningen på utvecklingsproblemen. Brooks har dock konsekvent avfärdat samtliga förslag och hävdar än idag att det saknas en generell lösning på strukturerad design av dataprogram.

Då spekulationer har funnits i att problem vid systemutveckling kanske mer är kopplade till utvecklingsmetodens angreppssätt en verktygets exakthet har ett flertal alternativ till utvecklingsprocesser kommit med åren. Den så kallade vattenfallsmodellen som först användes innebar att man delade upp arbetet i ett flertal faser. Då varje fas var färdig förflyttade man sig ”nedför vattenfallet till nästa steg”. På detta sätt fortlöpte arbetet till dess att systemet tagits i drift. Vattenfallsmodellen fick utstå ganska stark kritik för att den fastställde systemkraven mycket tidigt i utvecklingsarbetet och var oflexibel i det att det ej var möjligt att återkomma till en tidigare fas. En variant av vattenfallsmodellen utvecklades senare vilken nyttjade en iterativ modell, där det var möjligt att gå tillbaka till faser redan tidigare var avklarade. Detta innebar att det var möjligt att vid behov fokusera starkt på att konstruera standardiserade komponenter vid produktframställningen, för att därefter återgå till analysfasen. Trots sin iterativa ansats var dock fortfarande utvecklingsprocessen linjär och ej dynamisk i det att det ej var möjligt att arbeta i ett flertal faser parallellt.



**Figur 2-5 Boehms spiralmodell (Boehm [2])**

En välkänd modell som är utvecklad av Barry Boehm [2] är den så kallade spiralmodellen. Boehm inorporerade både den traditionella livscykelmodellen (vattenfallsprocessen) med prototyping och riskanalys i en avancerad,

evolutionistisk modell (se figur 2-5). I huvudsak kan man säga att utvecklingsarbetet fokuserar på fyra huvudaktiviteter; planering, riskanalys, utveckling och utvärdering. Modellen utnyttjar en iterativ ansats, där den första iterationen startar i mitten av cirkeln och sedan förflyttar sig utåt. Successiva iterationer följer då systemet blir mer och mer färdigutvecklat. Före det att en ny iteration påbörjas utförs en riskanalys, och då iterationen avslutas sker en sammanfattande utvärdering. Då modellen är oberoende av utvecklingsverktyg och validerar resultatet producerat vid varje iteration så har den många gånger blivit omnämnd som en tänkbar lösning på ”Software Crisis”. Modellen ser dock fortfarande på utvecklingen som separata produktionssteg och understödjer ej parallellt arbete vilket gör att ej heller den kan klassas som en helt dynamisk icke-linjär process.

### 2.2.3 Grupparbete

Då organisationer med tiden blivit mer och mer orienterade i arbetsgrupper (se Edgren et al [2]) är det rimligt att förutsätta att även användardeltagandet vid systemutveckling bedrivs i denna arbetsform. Då frågan hos systemutvecklare kanske mest är beroende av att så snabbt som möjligt få den information som behövs för att modellera verksamheten, finns det en risk att en arbetsgrupp kan verka trög och långsam om ej den kan presentera resultat i den takt som systemutvecklaren förutsätter.

En person som granskat detta noggrannare är Hägerfors [15] som presenterar tre fallstudier i systemutvecklingsarbete och arbetsgruppers tillvägagångssätt i detta. Hägerfors gör en koppling till Tuckman som beskrivit en grupps olika utvecklingsfaser; forming, storming, norming, performing och separation (se ibid. s 74). Fritt sammanfattat betyder det att då en arbetsgrupp sammanställts så gäller det initialt att bli accepterad som gruppmedlem (forming), varvid olika intressenter börjar hävda sig gentemot varandra (storming) och erkänna varandras status (norming). När gruppens deltagare till slut känner varandra, debatterar öppet och samarbetar och stöder varandra får medlemmarna en känsla av att arbetet kommer gå framåt och att hög kvalitet kommer att uppnås (performing). Hägerfors påpekar att övergången mellan faserna ej är linjär, och att det inom vissa grupper ej är möjligt att någonsin uppnå ett gott samarbete. Den sista fasen nämner Tuckman då detta vanligtvis är ett mycket känsloladdat ögonblick, som ofta påverkar gruppens medlemmar starkt.

Kunskapen om en arbetsgrupps utveckling och dess faser är en komplicerad vetenskap, då det enligt Hägerfors (ibid.) ofta är fallet att faserna går in i varandra. Då inget av de perspektiv Nurminen redovisar ser på arbetsgrupper som ett fenomen i organisationen, förefaller det viktigt att systemutvecklan-

de personal blir varse om att undermåligt resultat vid användardeltagande kan bero på en dåligt utvecklad arbetsgrupp.

## 2.3 Perspektiv och teorival

De flesta större organisationer är idag ej direkt verksamma inom den informationsteknologiska sektorn. Antalet organisationer med heltidsanställda systemerare, vars enda uppgift är att verka inom den egna organisationen är än mindre vanligt. Detta kan bero på att utvecklingen av den egna organisationen vanligtvis sker enligt en underförstådd hierarkisk struktur (se Grundén [11]). Nya organisationer satsar dock mycket på att kompetensutveckla sina medarbetare så att förändringsarbete kan bedrivas av ordinarie personal. Då personalen under förändringsanalysen funnit åtgärder som skulle behöva en datasystemteknisk implementation anlitas i sin tur externa systemutvecklare som försöker skapa sig sin bild av organisationen. Problem kan härmed uppkomma i det att personalen ej besitter produkttekniska kunskaper, och kan ha svårt att föreställa sig vad som är tekniskt möjligt att prestera. En annan lösning är att på konsultbasis anlita systemutvecklande personal som deltar vid personalens förändringsarbete och anpassar kravspecifikationen löpande för att kunna integreras och realiseras tillsammans med existerande system. En nackdel kan enligt Holmer/Karlsson [14] ses ur administrationens synvinkel även i detta tillvägagångssätt. Det faktum att "...företagen skaffat sig en mer direkt kontroll av den kunskapsbildning som är relevant för dem." (ibid. s 113) leder till att nyttjandet av extern arbetskraft kan leda till kunskap som i sin tur kan innebära merkostnader för organisationen (exempelvis kunskap om ISO-godkännande, konceptuella fel i befintliga datasystem och kunskap om kvalitetsnormer).

Jag anser att det är mycket svårt för ordinarie personal att utveckla sina kunskaper så pass långt att de är medvetna om utvecklingstekniska problem som kan uppkomma vid en eventuell datasystemutveckling. Systemutvecklingsmetoden som används för att sammanställa och klassificera åtgärderna bör vara strukturerad på ett sådant sätt att implementation alltid skall vara möjlig. Detta skulle även lösa de väsentliga problem som Brooks nämner [1] som de absolut allvarligaste vid datasystemutveckling. En lösning som används är att strikt styra upp de anställda genom att strukturera informationssystemet som en central knutpunkt. Då systemets behov och interaktionspunkter strikt specificerats innebär det mindre problem att definiera ett datasystem som överensstämmer med avbilden som tidigare gjorts.

Problem uppkommer vid nyttjandet av en grafisk modell som helt kretsar kring användaren, dess aktiviteter och informationsflöden. En systemut-

vecklingsmetod kallad arbetsanalys som är utvecklad av Grundén i samarbete med socialförvaltningen i Göteborg [11] nyttjar detta synsätt i en analysfas, utan direkt koppling till en senare teknisk implementation. För att en metod som är helt användarcentrerad i samtliga faser skulle kunna vara pappersbaserad tillsammans med ett oberoende vad gäller programspråk, skulle ytterligare en vy behöva skapas ur informationssystemet synvinkel. Detta för att kunna uppnå fullt dataverktygsberoende.

Jag upplever även att det för användare upplevs fördelaktigt om ett humanistiskt perspektiv nyttjas vid informationsanalysen så som Nurminen [24] beskriver det. Människor har lätt för att uppfatta sin arbetsituation ur en egocentrisk världsbild, där deras jag-medvetande sätts i fokus. Även kommunikationen är mycket lätt att beskriva då man använder en mer mänsklig betoning. Om man ber en anställd att beskriva sin vardag i arbetet så är det naturliga språket uppdelat på ens eget sociala kontaktnät och informationsmängder: ”Jag skapar denna typ av rapporter som jag sedan vidarebefordrar till person X. För att skapa denna rapport krävs det att informationen från person Y finns tillgänglig i tid.”. Sättet systemutvecklare generellt arbetar på skulle te sig ganska onormalt uttryckt som generell vokabulär: ”Jag är bearbetar rapport Z enligt process A1, efter det att jag som ingående informationsmängd fått en tidsstyrd mängd data från person Y. Denna rapport skickas sedan vidare inom informationssystemet till process A2.”.



### 3 Metodval och metodbeskrivning

Då syftet med rapporten är en sammanställning av synen på användare och användardeltagande på arbetsplatser och i litteratur inom Sverige, har en i huvudsak kvantitativ ansats valts. Med risk för att placeras i ett fack som grundforskare har jag tagit valet att helt utesluta egna mer praktiska studier och observationer. Då en kvalitativ analys av användardeltagandets tillämpning i de olika metoderna säkert hade varit både intressant och upplysningsrik, är det ändå sambandet mellan användardeltagande och resulterande system som ligger i fokus för sammanställningen. En kvalitativ analys av tillvägagångssättet riskerar att både bli oprecis och subjektiv, då de systemutvecklingsmetoder som granskas mest tjänar som riktlinjer och hjälpmedel för olika systemutvecklare.

Den litteratur som valts för den explorativa undersökningen har varit litteratur som framförallt redovisar praktiska fallstudier av införandet av databaserade informationssystem i olika organisationer. Detta motiverar min position till att sammanställningen är ”i huvudsak kvantitativ”. Det blir en kvantitativ sammanställning av ibland kvalitativa rapporter.

De systemutvecklingsmetoder som granskas har valts med tanke på de systemutvecklande företagens erfarenhet och storlek. Mindre systemutvecklande företag har färre resurser att lägga vid forskning om nya metoder, där större företag har särskilda avdelningar för just metodutveckling. Då tid går åt till kontakt med metodansvariga på respektive företag, tillsammans med ytterligare tid att få mig tillskickat metoderna begränsades antal konsultutvecklade metoder till två stycken, WM-Datas DELTA och Enators EFSOS.

Metoderna kommer ej att granskas i huvudsakligt komparativt syfte, eftersom målsättningen ej är att finna den metod som subjektivt mest verkar

främja en användarcentrerad utvecklingsansats. Istället kommer respektive metod att undersökas främst med hänsyn till synen på medarbetare och deras roll i systemutvecklingsarbetet.

## 4 Genomförande

### 4.1 Planering

Då en första kontakt med Enator visade att en ny utgåva av deras systemutvecklingsmetod EFSOS var under utveckling, innebar det att jag till en början helt kunde fokusera på att inhämta kunskap till rapportens teoretiska bakgrund och inledning. Initialt avsattes nio heltidsveckor åt läsande och sammanfattande strukturering, vilket skulle åtföljas av tre veckors metodstudie och resultatsammanställning. Detta innebar att jag hade mer än två och en halv månad på mig att få hem metoderna från de inblandade företagen.

### 4.2 Praktiskt arbete

#### 4.2.1 Litteraturstudie

Då det är mycket svårt att på förhand förutse vilken litteratur som passar bäst till studien utgick jag från de litteraturtips de involverade kontaktpersonerna bidrog med. Utifrån detta fåtal nyttjades sedan de referenser som tagits upp i litteraturen till att införskaffa vidare material, ner till maximalt tre nivåers djup. Den litteratur som presenterar praktiska fallstudier av effekterna på installerade datasystem beroende på utvecklingens tillvägagångssätt, hittades genom att via studentlitteratur beställa samtliga böcker tryckta efter 1990 som inkluderade sökorden systemutveckling och användardeltagande.

#### 4.2.2 Metodtillgänglighet och diskussion

Både Enator och WM-data var mycket restriktiva med att delge sin metod till en för dem okänd student. Då min rapports syfte till en början föreföll

ganska oklart då jag var anställd på skolan, kunde detta tolkas som att skolan i sig var intresserade av metoderna. En lösning till detta var ett skriftligt, signerat intyg av mig där jag specificerade att ej delge detaljerad information om någon metod utan tillstånd från metodutvecklaren.

En regelbunden diskussion har hållits med de involverade, och även om dessa samtal ej sammanställts i intervjuform så har den dock tillfört mycket idéer som varit nödvändiga för rapportens struktur. Diskussionen har även bidragit till ett flertal av de tankar jag sammanställer i sista kapitlet om vidare forskning.

Den person som visat sig mest otillgänglig är Göran Goldkuhl, vars metod VIBA helt inhämtats genom den litteratur som tidigare nyttjats som kursmaterial. Goldkuhl var ej heller intresserad av vidare diskussion kring ämnet, eller att delge någon information om sin kommande version av VIBA som för närvarande befinner sig under utveckling.

### **4.3 Problem**

Den tidsförskjutning som skedde av att Enators metod EFSOS skulle tryckas i ny upplaga ledde till att vårterminens examinationstillfälle missades och fick förflyttas till hösten. Då jag under våren även varit upptagen med att utveckla intranetsystemen på högskolan kom denna förflyttning av redovisningsdatum som en ganska välkommen tidsfrist.

De metodansvariga på WM-Data och Enator har båda en mycket hög kunskapsnivå. Mitt huvudsakliga problem uppstod då kommunikationen med dem ej fungerade tillfredsställande, mest på grund av min trots utbredd litteraturstudie mycket begränsade kunskap. Min definition av användarcentring som nyttjas i inledningskapitlet uppfattades ej likadant av någon av de involverade. När jag i min naivitet initialt skickade ut meddelande till de berörda där jag spekulerade i att användarcentrerade systemutvecklingsmetoder kunde upplevas som positivt av de anställda, fick jag svaret att båda företagen redan använde helt användarcentrerade metoder. Självklart var då min uppfattning om användarcentring ganska långt ifrån deras konkretisering.

På grund av rädsla för otillbörligt nyttjande av metoderna var jag tvungen att formulera och sammanställa ett avtal som fråntog mig rätten att duplicera och vidareförmedla detaljerad information om respektive metod. Detta gör att många avvägningar har genomförts i syfte att presentera metoden så ut-

förligt som möjligt, utan att delge några nyckelkomponenter som leder till att hela, eller delar av metoden kan återskapas.



# 5 Resultat

## 5.1 Insamlat material

Då synen på användare och deras eventuella deltagande i utvecklingsprocessen oftast styrs upp av vald metod, så ligger tyngdpunkten i detta kapitel på metodstudier. Då allt systemutvecklingsarbete startas av en process inom förändringsarbetet så är det naturligt att först redogöra för hur förändringsanalys bedrivs av respektive metod. Förändringsanalys är det ständigt pågående arbete som värnar om att ständigt förbättra kvalitén på arbetsplatsen. Detta tillvägagångssätt kommer samtidigt att sättas i perspektiv till de olika metodernas synsätt på kvalitet.

En granskning av akademiska forskningsstudier inom systemutveckling kommer även att redovisas, där en särskild granskning kommer att göras avseende de verk som utvärderar aktuellt forskningsområde i praktiska studier. Här redovisas de praktiska fallstudier som undersökt just användardeltagandet vid systemutvecklingsarbete och dess konsekvenser. Slutligen följer en detaljerad studie av de tre metoder som ingått i undersökningen, deras syn på användaren och organisationen samt grundläggande tillvägagångssätt.

### 5.1.1 Förändringsarbete och synen på kvalitet

Förändringsarbete utförs i syfte att förbättra och effektivisera verksamheten. Då ett externt konsultföretag anlitas så finns det en överhängande risk att ett datorbaserat informationssystem implementeras även om organisationen ej direkt kräver det. Detta märks tydligt i WM-Datas DELTA handbok [7] som ej innehåller någon förändringsanalys överhuvudtaget, och där konstruktion av konceptuella grafer börjar redan under första användarintervjun. Enators EFSOS [6] nämner att det kan vara intressant att utföra en förändringsanalys, men innehåller inte specifikationer om hur den bör bedrivas eller huruvida

den bör leda till implementation. Goldkuhl/Röstlingers innebörd av förändringsanalys [9] är att ”Den väsentliga poängen med FA är att inte initialt ta några förändringsåtgärder för givna.” (ibid. s 11). Enligt Goldkuhl [8] bör en systemutveckling enligt metoden VIBA alltid förekommas av en förändringsanalys, vilket leder till att systemutvecklingsarbetet är lika mycket verksamhetsutveckling som datasystemutveckling.

Det finns också skillnader i hur systemutvecklande personal ser på verksamheten och organisationens möjligheter till förbättring. Då de systemutvecklingsmetoder som granskats fokuserar på problem för att sedan lösa dessa, bygger Institutet för kvalitetsutveckling [16] sitt förändringsarbete på att finna möjligheter till tänkbara förbättringar. Dessa förbättringar utförs med vetskapen att det alltid kommer att uppstå problem vid förändringar, men i stället för att fokusera på problemen så vänder man sig till möjligheterna.

Då ovan nämnda systemutvecklingsmetoder sätter likhetstecken mellan begreppen kvalitet och systemkvalitet, utelämnas stora delar i verksamheten som för medarbetarna är det kanske mest viktiga; personlig livskvalitet. Grundén [11] är en av dem som starkast hävdar att systemutvecklande företag riskerar att förbise individens sociala kontaktnät och att studier av efterföljande implementation mycket sällan fokuserar på hur användares sociala situation förändras.

### **5.1.2 Användardeltagande i litteraturen**

En av de viktigare kvalitetsbestämmande faktorerna i ett databaserat informationssystem är dess användbarhet. Ett system som är komplicerat och svårt att anpassas för att passa in i användarnas vardag kommer också att ha en låg nyttjandefaktor<sup>5</sup>. Det finns ett flertal praktiska studier utförda inom detta område, det vill säga hur användarmedverkan nyttjas för att säkerställa kvalitén i den färdigställda produkten. En förklaring Lundeberg/Sundgren et. al. ger [23] är att det färdiga systemets flexibilitet kan försämrats beroende på bristande kommunikation mellan systembeställare och systemleverantör. Detta är även slutsatsen Lind [18] gör efter det att praktiskt arbetat med implementation av fyra systemutvecklingsprojekt, varav ett skett utomlands. Vid utvecklingsarbete i främmande land och kultur är det enligt Lind extra viktigt att välja gemensamma symboler att kommunicera med, då inte alltid språket fungerar som bästa budbärare av information. Då utvecklingen av Linds projekt skett efter varandra, har även metoden som nyttjats förfinats. I

---

<sup>5</sup> Se bland annat Lundeberg et. al [21] eller redovisningen av perspektiv i kap. 2.2.1

det senaste projektet valde Lind till exempel en mer objektorienterad ansats för att understödja icke-linjär utveckling.

Ett flertal böcker av den litteratur som insamlats var tänkt att nyttjas i en granskning av redovisade fallstudier. Detta med förhoppningen att författarna antingen skulle koppla sitt befintliga forskningsarbete till en praktisk studie av dess effekter, eller analysera hur en organisation skulle kunna utvecklas i och med införandet av lämpliga förändringsåtgärder. Den enda författare som bedrivit studier i akademiskt syfte med en praktisk efterföljande implementation är Lind [18] vars fallstudier av implementerade projekts kommunikationsutveckling bidragit mycket till rapportens resultat. Hägerfors [15] vars forskningsområde ligger på grupprocesser och på hur klimatet i en projektgrupp påverkar både arbetet och arbetets resultat har själv skapat tre fiktiva fallstudier utan koppling till verkliga organisationer. Att det rör sig om fiktiva studier avslöjar varken sammanställning eller bokomslag. Övriga hittills upptagna akademiska referenser (metoder eller metodutvecklare ej inräknade) har ingen koppling till praktisk verksamhet eller efterföljande realisation på något plan.

### **5.1.3 Systemutveckling ur olika perspektiv**

Då ett flertal forskare studerar användardeltagande som begrepp, torde ingen vara så erfaren som de metodutvecklare som praktiskt sett reaktioner och konsekvenser av metodförändringar genomförda över årens lopp. För att få ett bredare perspektiv på användaren ur systemutvecklarens ögon har två olika kategorier av metoder undersökts, där en är utvecklad i en akademisk forskningsgrupp och övriga skapade i konsultsyfte.

#### **VIBA**

VIBA står för verksamhets- och informationsbehovsanalys och är utvecklad i huvudsak av forskningsgruppen VITS vid Linköpings universitet. Utvecklingen är en följd av det arbete som bland annat Goldkuhl bedrivit i samarbete med ISAC-gruppen (se kapitel 2.1.1), och är mycket influerad av gruppens synsätt på användare och organisation. VIBA är en verksamhetsorienterad systemutvecklingsmetod som starkt genomlysas av synen på att datasystemens utveckling alltid innebär att verksamheten förändras, "Systemutveckling är verksamhetsutveckling." (Goldkuhl [8] s 23). Metoden bygger på principen att före en eventuell implementation av ett datasystem så föregås alltid arbetet med en förändringsanalys. Detta för att avgöra om behoven verkligen kräver en databaserad lösning för att tillmötesgå. Meto-

den diskuterar inte något praktiskt tillvägagångssätt för metoden, till exempel antalet användare som bör involveras, bildandet av projektgrupper eller kopplingen mellan systemutvecklad modell och implementerat system.

Likheterna med ISAC är som sagt övergripande, trots det flertalet år som passerat mellan de båda metodernas sammanställning. I-graferna som tidigare nämnts har utvecklats till att framtona individens roll i systemet genom de nya handlingsgraferna. Designen av graferna bygger dock fortfarande på konceptet av det totala informationssystemet. Initialt skapas alltid en avbild av verksamheten, och denna nyttjas sedan fram till dess att systemet är implementerat och i drift. Detta gör att medarbetares deltagande endast är intressant under analys och designfasen. Uppföljning av implementerade system, och dess konsekvenser på de sociala nätverk som existerar på arbetsplatsen omnämns inte alls.

Då medarbetaren skall beskriva sin arbetssituation för systemutvecklande personal så är endast ett fåtal informationsmängder beträffande arbetssituationen intressant. Medarbetarens situation sammanställs under rubriken ”Enhetsöversikt” där den anställdes yrkeskategori tilldelas en referenskod. För att överblicka användarens roll i det totala informationssystemet så anges här även i vilka grafer som användarens aktiviteter förekommer.

Då ett sociotekniskt synsätt tillämpas ses användare av systemet som mottagare och leverantörer av information. Begreppen ”Inanvändare” och ”Utanvändare” har skapats för att markera användarens roll i det totala systemet. För att markera att synsättet ej är systemteoretiskt nämns även att ”Information existerar inte för sig själv utan alltid i ett kommunikations-sammanhang.” (Goldkuhl [8] s 151).

## **DELTA**

WM-Datas DELTAMetod [7] är en konsultutvecklad systemutvecklingsmetod vars nyckelord är användbarhet. Metoden har akademiska kopplingar i det att den är utvecklad i samarbete med både Ericsson Infocom Consultants AB och Linköpings Universitet. Just beteckningen DELTA används för att förstärka bilden av ett öppet samarbete mellan användare, systemdesigners och tekniska kommunikatörer.

Likt VIBAMetoden så nyttjar även DELTAMetoden bilden av det totala informationssystemet som styrande enhet. De handlingsgrafer som används är nästan identiska med de handlingsgrafer som beskrivs av Goldkuhl [7], och tas i bruk efter det att användarna klassificerats i yrkeskategorier och ansvarsområden via enkäter. Till skillnad från VIBA så märks här en klar

förändring när det gäller anpassning av systemet till verksamheten och efterföljande tester. Användare deltar under analys- och designfasen då utförlig prototyptestning görs, tillsammans med efterföljande test av implementerat system.

DELTA förespråkar en tidig sammanställning av den designgrupp som kommer att styra utvecklingen. Antalet medarbetare som bör delta specificeras ej, men bör vara ett begränsat antal så att alla åsikter gör sig hörda. Metoden diskuterar ej heller design av de dokument som uppkommer under systemutvecklingsarbetet, då dessa situationsanpassas under arbetets gång.

DELTA innehåller ingenting som har med implementation att göra av den slutliga produkten. En prototyp skapas, men fortfarande kan en avbild komma att göras vid implementation i ett programspråk där prototyp och slutlig produkt skiljer sig åt. Inget användartest av systemet utförs när det väl befinner sig under utveckling. DELTAs huvudsakliga fokusering ligger på initial analys, där prototypingen spelar en stor roll. Användardeltagande bygger på en klassificering av användare via enkäter, där ett fåtal intressanta personer subjektivt väljs till intervju. Intervjun i sin tur leder till en design av grafer som bestämmer strukturen över systemet. Efter detta bestäms användbarhetsmål och en prototyp byggs. De utvalda användare som medverkar i systemutvecklingsarbetet ingår i samtliga dessa faser.

### **EFSOS, PAS och PTM**

Enators system för verksamhetsutveckling består av ett flertal komponenter. De tre jag valt att fokusera på kallas för EFSOS, PAS och PTM. EFSOS är Enators metod för objektorienterad systemutveckling, PAS står för praktisk användarmedverkan vid systemutveckling och PTM står för praktisk testmetod. De olika metoderna nyttjas parallellt, dock härrör all systemutveckling från EFSOS.

EFSOS [6] skiljer sig från de flesta systemutvecklingsmetoder i det att den är helt objektorienterad. Den är dynamisk och enkelt föränderlig, då samtliga aspekter som bidragit till objektorienteringens utbredning anammats. Detta innebär också att ett flertal personer kan arbeta parallellt med metoden, i olika utvecklingsfaser. Systemet som skapas ses som ett flertal komponenter, vars kommunikationssätt och innehåll definieras då tillräckligt med information framkommit för att fastställa detta. Användare redovisas ej i modellgraferna, utan definieras enligt ett separat system liknande enhetsöversikten nyttjad av Goldkuhl.

På grund av just dess dynamiska struktur, tillåter EFSOS sina systemutvecklare att bygga användargränssnittet i samma modell som verksamhetsstrukturen. Detta innebär att behovet av en separat prototyp helt faller bort. Prototypen och den färdiga applikationen är samma system. Målet med detta är att minimera antalet avbildningar (av enator kallat transformationer), vilket också syns vid användandet av både objektorienterade databaser och utvecklingsverktyg. Samma modell kan användas från analys och design till färdig produkt, och Enator förklarar valet av angreppssätt enligt:

”Den viktigaste förutsättningen för att uppnå denna symbios mellan verksamhetsutveckling och systemutveckling är att anamma en teknologi där *modeller* av verksamheten enkelt kan avbildas i företagets IT-system. Först då kan man skapa system som snabbt tar hand om och omsätter förändrade behov och inriktningar i verksamheten.” (EFSOS [6], s 49).

Då användarmedverkan anses vara en övergripande faktor vid systemutvecklingsarbete har Enator valt att skapa en särskild metod för just detta, kallad PAS. Denna metod [4] förespråkar bildandet av en arbetsgrupp om 3-10 personer inklusive användare, där användbarhet är den viktigaste faktorn för gruppens arbete. Detta arbete sker parallellt med det att systemutvecklingsarbetet fortgår med hjälp av EFSOS, och kan ses som länken mellan systemutvecklande personal och dess användare. PAS beskriver både tillmötesgående mot användarna, tillsammans med detaljerad information om hur sociala aspekter och synpunkter skall vägas in i arbetet. Utifrån den modell som byggs av arbetsgruppen skapas sedan den prototyp som slutligen används för den färdiga produktens utseende. Tänkbara framtida arbetsuppgifter skapas av gruppens deltagare och påverkar sedan hela organisationen.

För att verifiera kvalitetsnivån i det färdiga systemet nyttjas sedan PTM, praktisk testmetod [5]. Detta arbete beskrivs mycket omfattande och innehåller ett flertal tester som ställs i förhållande med de kvalitetskrav som ställs på produkten. Ju mindre acceptans för fel hos användaren, desto större testarbete blir involverat i utvecklingsarbetet. Arbetet sker i isolerat miljö, då endast oväsentliga fel skall hittas (se Brooks kap. 2.2.2). Enator uttrycker detta som att ”Man kan aldrig testa fram bra kvalitet. Kvalitet måste byggas in i produkten från början.” (ibid. s 10).

## 5.2 Resultatanalys

### **5.2.1 Förändringsarbete och synen på kvalitet**

Då en systemutvecklingsmetod framställs så tilldelas den ofta ett syfte. Detta syfte varierar beroende på utvecklingsmiljö och de personer som är involverade i metodframställningen. Är det ett konsultföretag som utvecklat metoden, vars målsättning med verksamheten är att skapa så många kvalitativa och välfungerande system som möjligt, är det naturligt att metoden syftar till att lösa problem som utvecklingshastighet, fel i struktur och programkod samt effektiv användaracceptans. Om detta löses genom en hög grad användardeltagande är ej relevant, det viktigaste är att finna en lösning på problemen. Om metoden å andra sidan är utvecklad i isolerad miljö, där problem relaterade till praktiska händelser ej uppkommer finns det en risk att dessa faktorer förbises. Fördelen kan däremot vara att andra faktorer än den egna verksamhetens lönsamhet kan hamna i fokus.

Ett tydligt exempel är det faktum att ingen av de konsultutvecklade metoder som granskats ifrågasätter utvecklingen av ett datasystem. Detta kan tyckas självklart då det ligger ekonomiska aspekter bakom systemutvecklingsarbetet, men det finns även andra tänkbara orsaker bakom detta. Om en efterföljande implementation ej sker så skulle det systemutvecklande företaget få förklara varför de valt att undvika denna process. Detta skulle kunna tolkas som om systemutvecklande personal ej hittat tillräckligt antal problem som kan förbättras, vilket leder diskussionen till om något annat systemutvecklande företag kunnat utföra arbetet effektivare.

Det faktum att samtliga metoder bygger på principen att finna problem för att sedan lösa dessa, kan bero på att det blir mycket enkelt att sammanställa en kvantitativ resultatsammanställning över hur organisationen kan förbättras. Om förändringsanalysen sker med hjälp av egen personal kanske detta inte är lika relevant, då en fokusering på att förbättra existerande metoder kan ske (exempelvis de metoder som Landstinget nyttjar i QUL [17]). Personalens eventuella brist på kunskaper i datasystemutveckling och brist på tid kan även detta vara bidragande orsaker till upplägget.

### **5.2.2 Användardeltagande i litteraturen**

Då ingen kommunikation föreligger mellan beställare och leverantör av ett datasystem så kommer produkten antagligen att behöva specialanpassas för att passa in i verksamheten. Ett bra exempel på detta är alla de standardprogramvaror med ordbehandling och kalkyl som finns att inhandla. Dessa produkter innehåller möjligheten att anpassa både utseende och funktion hos produkten så att ett standardiserat gränssnitt är möjligt att uppnå inom orga-

nisationen. Om denna anpassning skall vara genomförd i det att produkten tas i drift krävs det ett samarbete där medarbetarnas arbetsituation tjänstgör som grund för funktionaliteten och strukturen i systemet. Då personer med ofta vitt skilda tekniska bakgrunder skall samarbeta mot något så pass avancerat som arbetssituationsanalys och strukturdesign av ett datasystem, kommer kommunikationsproblem att uppstå.

Då det föreligger risk för att kommunikationen i sin tur kan påverka kvalitén i implementerade system är det anmärkningsvärt att inte fler författare bidragit till praktiska studier. Jag menar i detta ej studier som lett fram till konstruktionen av en ny modell eller metod, utan studier som utförts i syfte att förstärka befintliga teorier. Ett bra exempel på detta är Lind [18] som parallellt med forskningsarbetet utförde ett flertal fallstudier för att granska effekterna av de nya idéerna.

### **5.2.3 Systemutveckling ur olika perspektiv**

Under en förändringsprocess som påverkar samtliga anställda så torde det vara en självklarhet att samtliga medarbetare har något att säga till om vid en tänkbar framtida utveckling. Institutet för kvalitetsutveckling [16] är några av de som förespråkar en total deltagandekvot där samtliga medarbetare deltagar. Detta fungerar troligtvis bra om förändringsarbetet sker i den egna miljön, av samma personal som får nöjet att uppleva förändringarna. I det fallet då man ej är säker på utgången (eventuellt personalbortfall) eller då misstanke finns att tekniska system bör implementeras kan det dock synas intressant att involvera externa konsulter. Skillnaden i utvecklingsarbete är då att en majoritet av de anställda ej längre har möjlighet att påverka sin egen arbetsplats utveckling, främst på grund av att mindre utvecklingsgrupper bildas. Fördelen till detta syns tydligt om man granskar de problem som uppkommer vid arbete i grupp (se kap 2.2.3), och vetskapen om att problemen förstärks då gruppens storlek ökar. Detta bör även sättas i perspektiv till de kommunikationsproblem som tidigare nämnts, vilket försvårar arbetet avsevärt ju större antal personer som deltagar. Validering av att samtliga anställda förstått innebörden i det budskap som förmedlas är nästan omöjligt i en större organisation. Utöver detta så finns det även ett flertal personer som inte alls är intresserade av att deltaga i utvecklingsarbetet överhuvudtaget.

Att det är nödvändigt att försöka skapa en avbild av verksamheten omtvistas ej i någon metod, utan det är tillvägagångssättet som skiljer sig åt. Då VIBA och DELTA har en relativt linjär utvecklingsprocess med mycket redundant information blir ett iterativt arbete ofta omständligt, och kan i värsta fall tillföra mer fel än de som ursprungligen avsågs att korrigeras. En lösning på

detta har Enator i sin metod EFSOS, då objektorienteringen med alla dess fördelar leder till att minimal redundans uppstår. Nyttjandet av arv via bas-klasser tillsammans med möjligheten att arbeta dynamiskt med ett flertal olika vyer av samma data, leder till möjligheter som att kunna arbeta gemensamt i grupp mot det totala informationssystemet samtidigt.

Objektorienteringen i EFSOS leder också till att utvecklingen av en *separat* prototyp som förespråkas i DELTA ej längre förefaller relevant. Det är endast att definiera vilka objekt (verksamhets- och användargränssnittskomponenter) som skall ingå i prototypen och påbörja testerna. Detta innebär samtidigt att den teknik som förfinats i prototypen är direkt användningsbar i den färdiga produkten, som endast har ett utökat antal komponenter och kopplingar. Då objektorienterade utvecklingsverktyg är en ganska ny företeelse, så förefaller det ganska klart varför varken VIBA eller DELTA har en starkare koppling till implementationsfasen. Enators metod EFSOS är så pass ny att denna koppling ej hade varit möjlig att genomföra för bara några år sedan, just på grund av bristen på verktyg.

VIBAmetoden diskuterar varken test av produkten innan implementation inleds eller uppföljning och säkerställande av kvalitén. Detta kan bero på det faktum att den är utvecklad helt i en isolerad miljö som diskuterats innan. Intressekonflikter som kan uppstå på arbetsplatser vid brister i tillvägagångssättet är svåra att förutse utan praktisk tillämpning.

Vid en eventuell organisationsförändring så är det ett fåtal människor som ej tänker egocentriskt. Möjligheten att avsäga sig sin tjänst då den totala lönsamheten kan ökas vid införandet av ny teknik är ett beslut som kan vara svårt att fatta. Då det i samtliga metoders tillvägagångssätt bildas mindre arbetsgrupper där ett fåtal anställda medarbetare tas med, blir det en omöjlighet en att undvika subjektivitet i väsentliga frågor gällande arbetsfördelning mellan människa och maskin. Dessa aspekter diskuteras inte enligt metoderna i fråga, men påverkar säkerligen besluten (se Grundén [11] för en vidare diskussion angående detta). Det faktum att en människas arbetsuppgifters sammanhang åskådliggörs i separata handlingsgrafer gör att det är svårt för den enskilde medarbetaren att granska om gruppen korrekt avbildat dennes uppgifter. Detta kan i sig vara skäl nog att begränsa gruppens antal deltagare.



## 6 Slutsatser

Rapporten har behandlat följande frågor:

- Bör användare delta vid systemutvecklingsarbete, och i så fall varför?
- Vilka fördelar respektive nackdelar finns det i att låta användare delta i arbetet?
- Inom vilka faser i utvecklingsarbetet bör man låta användare delta?
- Hur organiseras processarbetet?

Min grundläggande ansats i denna rapport baserade sig helt på den begränsade syn av användare som nyttjats i verksamhetsmodellering enligt VI-BA/SIMM-metoden. Efter genomgången resultatsammanställning fann jag det mycket positivt att se vilken utbredning diskussionen om användares roll i systemutvecklingsarbetet har fått med tiden. Frågan om användare bör delta vid systemutvecklingsarbete är numera ett självklart ja, då ett verksamhetsutvecklat system kräver en respons från användare för att kontinuerligt uppdatera systemutvecklaren med dagsaktuell information om hur arbetsituationen förändras. Denna kommunikation främjar även kvalitén i det färdiga systemet då minimalt antal förändringar kommer att behövas i slutprodukten, och skulle i så fall även hjälpa till att minska underhållstiden på produkten (se kap. 2.2.2 för en diskussion om utvecklingsproblematik). Även om synsättet på användare enligt Nurminen förändrats med årens lopp så är det ej att förglömma att ett flertal större organisationer fortfarande bygger på äldre datasystem enligt den gamla skolan, vilket kan leda till intressekonflikter med nya perspektiv.

Även om ett system skapas som passar perfekt med organisationen, så tjänar dess syfte inte något till om produkten ej nyttjas av de anställda. Användaracceptans är den kanske starkaste orsaken till att så många systemutvecklare förespråkar användardeltagande i utvecklingsprocessen, då en känsla av personligt inflytande i slutsystemet infinner sig hos medarbetarna.

Det är icke att förglömma att även avdelningschefer och verksamhetsadministrativ personal räknas som användare av systemet, och kan bidra med mycket viktig information om organisationsstruktur och tidigare verksamhetsutveckling.

Självklart finns det även nackdelar inom användardeltagandet, framför allt de som drabbar den systemutvecklande personalen. Sett endast ur utvecklingsteknisk synpunkt så är det naturligt att programmeraren föredrar ett minimalt antal störmoment i utvecklingsarbetet, även om dessa bidrar till att öka den generella kvalitén. Vid nyttjandet av metoder som förespråkar redundant data tillför små synpunkter oftast enorma förändringar (se även här kap. 2.2.2 om utvecklingstekniska problem). Medarbetarnas arbetsituation kan även drabbas i det att deras arbetsrutin förändras då de regelbundet måste delta i utvecklingsgruppens arbete. Hårt tidspressade organisationer med korta deadlines kan ha svårt att bidra med omfattande personal då omsättningen skulle minska markant.

Då granskade metoder använder olika angreppssätt för att konstruera systemet (både linjära och parallella utvecklingsprocesser förekommer) så är det vid en idealtypisk granskning ej möjligt att utvärdera huruvida användardeltagandet praktiskt får betydelse för slutprodukten. I det fall då det via objektorienterad utveckling är möjligt att arbeta parallellt finns det i alla fall möjlighet att låta användarna delta under hela processarbetet. I det fall då en objektorienterad ansats ej nyttjas är det dock trots allt möjligt att via iterativt upprepande processer regelbundet få med användarnas åsikter i arbetet, då riskanalys och prototypstest är en ständigt återkommande aktivitet (se Boehms spiralmodell kap. 2.2.2).

Då arbete i isolerade projektgrupper förekommer i samtliga redovisade metoder används användardeltagandet endast under analys- och designfasen. Då dessa är iterativa är det dock möjligt att återgå till dessa faser under implementationsfasen, dock så förstärks komplexiteten i systemet enormt om ansatsen ej är objektorienterad. DELTA och EFSOS förespråkar även en stor del användardeltagande vid efterföljande test av implementerat system för att granska hur väl systemet passar in i organisationen.

Det är slutligen även intressant att se hur det selektiva urvalet som bidrar till bildandet av arbetsgruppen sker. Då deltagarnas lämplighet avgörs genom enkätsammanställning anser tydligen samtliga metoder att kreativitet och förmåga att arbeta i grupp kan mätas genom en skriftlig intervju. Arbetsgruppens storlek diskuteras inte enligt VIBAMetoden, men sätts i DELTA och EFSOS till ett begränsat antal, helst under tio personer.

## 6.1 Generell metodutvärdering

Då min kunskapsnivå inom området från början endast var relaterad till de kurser jag läst under min studietid, var det ett steg i rätt riktning med en bred, explorativ ansats. Den kunskap som inhämtades via litteraturen var till stor nytta när det blev dags att granska utvalda metoder. Även vid kontakt med involverade metodansvariga upplevdes det som positivt att kunskapsnivån förhöll sig högre än den genomsnittlige studenten.

Då studien skulle involvera praktiska fallstudier, inhämtades litteratur som enligt specifikation skulle redovisa ett flertal situationer där det aktuella forskningsområdet skulle praktiseras. Olyckligtvis var det endast en författare kunde redovisa resultat som direkt var kopplat till aktuellt forskningsområde. En fallstudie utförd på eget bevåg skulle troligtvis ej berika innehållet ytterligare, då metodernas tillämpning relativt strikt styrs upp av metoden. Det som dock skulle kunna tillföra ytterligare värde hade varit en intervju med anställda i olika organisationer som bedriver förändringsarbete. Detta för att få med användarnas uppfattning och reaktioner mot konsulter inverkan i organisationen.

## 6.2 Sammanfattande utvärdering av arbetet

Jag upplever att resultatet av rapporten är något som kan nyttjas av både lärare och studenter inom systemvetenskapliga områden. Då kunskapen om nya tekniker hålls strikt konfidentiell inom respektive företag är vetskapen om dessa tillvägagångssätt en mycket sällsynt kunskap. För att lärare skall kunna tillgodogöra sig kunskapen om ett effektivt utvecklingssätt med objektorienterade systemutvecklingsmetoder krävs både kunskap om verksamhetsmodellering och objektorienterad programmering, tillsammans med en förståelse av de sociala aspekter som uppkommer vid förändringsarbete. Utöver detta så krävs det även goda kunskaper i relationella och objektorienterade databaser samt tänkbara kopplingar mellan dem och övrigt system. All denna kunskap behövs dock inte för att förstå att objektorienterade systemutvecklingsmetoder är i stark frammarsch inom företagsvärlden, och kommer troligtvis förr eller senare att även influera undervisningssättet i högskolor runt om i landet.

Då jag som student i våras inte var medveten om att användare kunde delta i utvecklingsarbetet inom någon fas, visar detta bara vilken enorm skillnad i perspektiv mellan konsultföretag och akademiska institutioner som föreligger.

### 6.3 Egna reflexioner

För nästan trettio år sedan, 1969 framställde ordföranden i Standardkommittén för Administrativ teknik och databehandling ett förslag att SIS<sup>6</sup> skulle verka för enhetlighet inom utveckling och dokumentation av ADB-system. Ett större samarbete med näringsliv och statlig verksamhet inleddes, och resulterade tio år senare i handboken: Riktlinjer för administrativ systemutveckling (RAS) [25]. Modellen har varit väl använd av olika större företag, men då under namn som var direkt kopplade till verksamheten (se Grundén [11] s 76). SIS-RAS var ettappindelad och icke-iterativ, och det som framför allt gjort modellen känd är att användarna av systemet ej tilläts delta i utvecklingsarbetet förrän projektet togs i drift. De anställda sågs ej som en del av verksamheten förrän de skulle lära sig datasystemet, och metoden var starkt influerad av det systemteoretiska perspektivet.

Utvecklingen har sedan dess fortskridit framåt i relativt snabb takt, och samtliga metodutvecklare talar idag om användbarhetsfaktorer och användardeltagande i systemutveckling. Det faktum att användare har mer att säga till om i vissa metoder, kan endast ses som att övriga metoder ej är lika långt framåtskridna. Nurminen [24] beskriver att en övergång till ett mer humanistiskt perspektiv håller på att ske, med användaren i centrum. Detta innebär en total omvälvning av de metoder som arbetar systemteoretiskt eller socio-tekniskt. Ofta blir paradigmskiftet så stort att nya metoder måste skapas för att ersätta de äldre.

Kritik som bland andra Grundén [11] riktar mot att de sociala aspekterna ej inkluderas i systemutvecklingsarbetet är delvis korrekt än idag. Det företag som idag har utvecklat en metod som går att nyttja helt användarcentrerat är Enator, vilka har specificerat användardeltagandets tillvägagångssätt i sin separata metod PAS. Denna fokuserar på psykosociala aspekter liksom arbetslivskvalitet. Det som fortfarande ej diskuteras är datasystemens påverkan på sociala kontaktnät och arbetsuppgiftsfördelning inom organisationerna.

För att vidareutveckla de systemutvecklingsmetoder som idag används inom ett flertal högskolor i Sverige behövs en utveckling av kompetensen bland lärarna. De objektorienterade ansatser som nyttjas av industrin kräver att systemutvecklaren behärskar både objektorienterad verksamhetsmodellering, programspråk och databashantering då terminologin är identisk över alla faser. Eftersom det finns brist på akademiska metoder som baseras på denna teknik kommer även utveckling av framtida metoder att behöva ske. För att

---

<sup>6</sup> SIS står för Standardiseringskommissionen i Sverige

dessa skall vara praktiskt användbara kommer ett utökat samarbete att krävas mellan olika institutioner för att bemästra både de tekniska och sociala aspekterna vid förändringsarbete och produktutveckling.

## 6.4 Idéer för fortsatt forskning

Då systemutvecklingsarbete fortfarande bedrivs som en isolerad process relativt avskärmad från den organisation där systemet skall tas i drift, kan det finnas ett behov av en ökad öppenhet. En tänkbar utveckling bland systemutvecklingsmetoder är en övergång till online-baserade system där olika vyer kan sättas på tillgänglig information för att ändra åtkomligheten. Detta skulle innebära att distansarbete för systemutvecklare skulle kunna realiseras mer effektivt, och pappersmässig dokumentation skulle minska radikalt.

Om man ser på de metoder som undersökts så understödjer samtliga någon form av CASE (Computer Assisted Software Engineering) verktyg som effektivt minskar behovet av tryckt dokumentation. Generellt sett kan man säga att metoder som arbetar enligt inkapslingstänkande (till exempel objektorienterade ansatser) som bygger informationsstrukturen i ett flertal abstraktionsskikt har mindre behov av ett CASE verktyg då modellering sker i ett flertal separata komponenter. I det fallet man arbetar enligt principen ”botten-upp” vilket först identifierar den totala informationsstrukturen är det mycket omständligt att tillföra nya komponenter då ett flertal förändringar sker vid varje tillagd komponent (se kap. 2.2.2). Ett CASE-verktyg som stödjer hela utvecklingsprocessen och dessutom är objektorienterat skulle radikalt underlätta att implementationsmässiga misstag uppkommer. Frågan om ett sådant verktyg tillsammans med en effektiv modell för utvecklingsprocessen (exempelvis Boehms spiralmodell kap. 2.2.2) skulle underlätta de problem som Brooks (se även här kap 2.2.2) uttrycker som väsentliga återstår att se. De verktyg som idag finns tillgängliga (Obsydian m.fl.) är så pass nya att deras effekt på systemutvecklingens korrekthet fortfarande är okänd.



# Referenslitteratur

1. F. Brooks: *No silver bullet – essence and accident in software engineering*, IEEE computer, vol. 20 nr. 4, April 1987
2. B. Boehm: *A Spiral Model of Software Development and Enhancement*, IEEE Computer, May 1988, p. 61-72
3. J. Edgren: *Lära på jobbet*, Svenska arbetsgivareföreningen, 1990
4. Enator AB: *Praktisk Användarmedverkan vid Systemutveckling (PAS 2.0)*, Enator informationssystem AB, Kista, 1997-05-30
5. Enator AB: *Praktisk Testmetod (PTM 2.1)*, Enator informationssystem AB, Kista, 1997-05-30
6. Enator AB: *Objektorienterad utveckling (EFSOS 3.0)*, Enator informationssystem AB, Kista, 1996
7. Ericsson Infocom Consultants: *DELTA Method Handbook*, Ericsson Infocom Consultants AB, Linköping, 1994
8. G. Goldkuhl: *Verksamhetsutveckla datasystem*, Affärlitteratur AB, Linköping, 1993
9. G. Goldkuhl / A. Röstlinger: *Förändringsanalys*, Studentlitteratur, 1988
10. K. Grundén: *Ett människoorienterat perspektiv på systemutveckling*, MDA-rapport 1992:4
11. K. Grundén: *Människa Organisation ADB-system*, Studentlitteratur, 1992
12. K. Grundén: *Redovisning av en intervjuundersökning om ADB-systemutveckling*, Göteborgs universitet, 1986
13. H. Göranson / I. Ottersten: *Objektorienterad utveckling med COOL-metoden*, Studentlitteratur, 1993
14. J. Holmer / J. Karlsson: *Kvalifikation*, Konsultförlaget AB, 1991
15. A. Hägerfors: *Att samlära i systemdesign*, Studentlitteratur, 1995

16. Institutet för kvalitetsutveckling (SIQ): *Utmärkelsen svensk kvalitet*, Institutet för kvalitetsutveckling, 1997
17. Landstingsförbundet: *Kriterier och anvisningar för Qvalitet Utveckling Ledarskap*, Landstingsförbundet, 1996
18. B. Lind: *Communication in system development*, HB Limes, Viskafors, 1991
19. B. Langefors: *Theoretical analysis of information systems*, Studentlitteratur, Lund, 1966
20. B. Langefors: *Introduktion till informationsbehandling*, Natur och kultur, Stockholm, 1971
21. M. Lundeberg / G. Goldkuhl / A. Nilsson: *Systemering*, Studentlitteratur, 1979
22. M. Lundeberg / E. Andersen: *Systemering – informationsanalys*, Studentlitteratur, 1975
23. M. Lundeberg / B. Sundgren: *Att föra verksamheten framåt*, Studentlitteratur, 1996
24. M. Nurminen: *People or Computers: Three ways of looking at information systems*, Studentlitteratur, 1988
25. SIS: *Riktlinjer för administrativ systemutveckling*, Minab, Surte, 1979
26. F. W. Taylor: *Rationell verkstadsledning enligt taylorssystemet*, Sveriges Industriförbund, Stockholm, 1923



## Högskolan i Borås

Högskolan i Borås bildades i samband med högskolereformen 1977. Högskolans textila utbildningar har dock sitt ursprung redan i den Tekniska Väfskolan från 1866.

I dag är Högskolan i Borås en nationell högskola med studenter från hela landet. Högskolan i Borås ger utbildningar inom följande ämnesområden; biblioteks- och informationsvetenskap, textil, teknik, barn- och ungdomspedagogik samt data och ekonomi.

Forskning och utvecklingsarbete är en expanderande del av högskolans verksamhet. En viktig del i detta arbete är en nära samverkan med företag, statliga myndigheter och kommuner både i Sverige och utomlands.

## Institutionen för data- och affärsvetenskap

Institutionen för data- och affärsvetenskap ger utbildningar inom följande tre områden: ekonomi, informatik och språk.

Borås är en handelsstad och ekonomutbildningar med specialinriktningar har därför utvecklats naturligt. Flera av utbildningsprogrammen som ges inom ekonomi är de enda i sitt slag i landet.

Institutionen för data- och affärsvetenskap ger följande utbildningar: Ekonomutbildning 120/160 poäng, Ekonomutbildning med inriktning mot direktmarknadsföring/postorder 120/160 poäng, Dataekonomutbildning 140/160 poäng samt Systemarkitektutbildning 120/180 poäng.

Därutöver erbjuder institutionen fristående kurser i bland annat språk och ett antal humanistiska ämnen s.k. Liberal Artskurser.

### För mer information:

Högskolan i Borås  
501 90 Borås  
Tel 033-16 40 00  
Fax 033-16 40 03  
E-post: [Info@hb.se](mailto:Info@hb.se)  
Internet: <http://www.hb.se>