

**Användarmedverkan i traditionella  
systemutvecklingsmetoder**

**(HS-IDA-EA-01-320)**

**Susanna Lundin (a98suslu@student.his.se)**

*Institutionen för datavetenskap*

*Högskolan i Skövde, Box 408*

*S-54128 Skövde, SWEDEN*

Examensarbete på det systemvetenskapliga programmet under  
vårterminen 2001.

Handledare: Beatrice Ahlenjung

## **Användarmedverkan i traditionella systemutvecklingsmetoder**

Examensrapport inlämnad av [Susanna Lundin] till Högskolan i Skövde, för Kandidatexamen (B.Sc.) vid Institutionen för Datavetenskap.

**2001-06-08**

Härmed intygas att allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och att inget material är inkluderat som tidigare använts för erhållande av annan examen.

Signerat: \_\_\_\_\_

## Användarmedverkan i traditionella systemutvecklingsmetoder

Susanna Lundin (a98suslu@student.his.se)

### Sammanfattning

De problem som följer den traditionella systemutvecklingen är att det är många informationssystem som inte blir implementerade eller använda på grund av att användarna inte accepterat det nya informationssystemet, likaså är ofta projekten överbudgeterade och tidsöverskridande. En av orsakerna till detta är att användarna inte deltar aktivt under utvecklingsprocessen, vilket gör att systemutvecklarna inte får all den information de behöver för att utveckla ett informationssystem som motsvarar förväntningarna.

I detta examensarbete undersöks om tillvägagångssättet participatory design (PD), som innebär att användaren aktivt deltar i hela utvecklingsprocessen, går att integrera med den traditionella systemutvecklingsansatsen, där utvecklingen sker med användarna som stöd till systemutvecklarna.

Integrationen har gjorts genom att dels byta ut tekniker i systemutvecklingsmetoden SSADM och dels genom att kombinera SSADMs tekniker med PD-tekniker. Detta resultat har används tillsammans med intervjuer för att avgöra om PD-tekniker kan användas i alla traditionella systemutvecklingsmetoder. Resultatet av integrationen visar att det är möjligt att integrera PD-tekniker med den traditionella systemutvecklingsansatsen.

**Nyckelord:** Participatory Design, Användarmedverkan, SSADM, Traditionell systemutveckling, Framtidsseminarium, Kooperativ prototyping

# Innehållsförteckning

<b>1</b>	<b>INTRODUKTION .....</b>	<b>1</b>
<b>2</b>	<b>BAKGRUND.....</b>	<b>3</b>
2.1	Utvecklingsmetoder.....	4
2.2	Systemutvecklingens olika ansatser.....	5
2.3	Skillnader mellan systemutvecklingsansatserna och den traditionella ansatsen.....	5
2.4	Den traditionella systemutvecklingens livscykel .....	7
2.5	Participatory design.....	10
2.6	Tekniker för participatory design .....	12
2.6.1	Framtidsseminarier .....	13
2.6.2	Design genom rollspel .....	14
2.6.3	Mock-up design.....	15
2.6.4	Kooperativ prototyping .....	16
2.6.5	Kontextuell utfrågning.....	17
<b>3</b>	<b>PROBLEMSTÄLLNING .....</b>	<b>19</b>
3.1	Examensarbetets frågeställningar .....	19
3.2	Avgränsning .....	20
3.3	Förväntade resultat.....	20
<b>4</b>	<b>METOD.....</b>	<b>21</b>
4.1	Möjliga metoder.....	21
4.1.1	Intervjuer .....	21
4.1.2	Fallstudie.....	21
4.1.3	Litteraturstudie.....	22
4.2	Val av metod .....	22
4.3	Plan för arbetet .....	23
<b>5</b>	<b>PARTICIPATORY DESIGN .....</b>	<b>26</b>
5.1	Framtidsseminarium.....	26
5.1.1	Den kritiska fasen .....	26
5.1.2	Fantasifasen.....	27

5.1.3	Implementationsfasen .....	27
5.1.4	Fördelar och nackdelar .....	27
<b>5.2</b>	<b>Kooperativ prototyping .....</b>	<b>28</b>
5.2.1	Fördelar och nackdelar .....	28
<b>6</b>	<b>SYSTEMUTVECKLINGSMETODEN SSADM.....</b>	<b>30</b>
<b>6.1</b>	<b>Tekniker i SSADM.....</b>	<b>31</b>
6.1.1	Kravdefinition.....	32
6.1.2	Dialogdesign .....	32
6.1.3	Dataflödesmodellering .....	32
6.1.4	Logisk datamodellering .....	33
6.1.5	Alternativa affärssystem.....	33
6.1.6	Funktionsdefinition.....	33
6.1.7	Relationsdataanalys .....	33
6.1.8	Alternativa tekniska system.....	33
<b>6.2</b>	<b>Faser i SSADM.....</b>	<b>34</b>
6.2.1	Undersökning av möjligheter .....	34
6.2.2	Kravanalys.....	36
6.2.3	Kravspecifikation.....	40
6.2.4	Logisk systemspecifikation .....	42
6.2.5	Fysisk design.....	45
<b>6.3</b>	<b>Användarmedverkan i SSADM.....</b>	<b>48</b>
<b>6.4</b>	<b>Sammanfattning av SSADM .....</b>	<b>49</b>
<b>7</b>	<b>INTEGRATION AV PD OCH SSADM.....</b>	<b>51</b>
<b>7.1</b>	<b>Fas 0 - möjligheter .....</b>	<b>51</b>
<b>7.2</b>	<b>Fas 1 - undersökning av nuvarande omgivning.....</b>	<b>52</b>
<b>7.3</b>	<b>Fas 2 - alternativa affärssystem .....</b>	<b>53</b>
<b>7.4</b>	<b>Fas 3 - definition av krav .....</b>	<b>54</b>
<b>7.5</b>	<b>Fas 4 - val av alternativa tekniska system.....</b>	<b>54</b>
<b>7.6</b>	<b>Fas 5 - logisk design och fas 6 - fysisk design .....</b>	<b>55</b>
<b>7.7</b>	<b>Sammanfattning.....</b>	<b>55</b>
<b>8</b>	<b>GENOMFÖRANDE AV INTERVJU .....</b>	<b>56</b>
<b>9</b>	<b>RESULTAT FRÅN INTERVJUN .....</b>	<b>58</b>
<b>10</b>	<b>SLUTSATS .....</b>	<b>60</b>

<b>11</b>	<b>DISKUSSION .....</b>	<b>61</b>
<b>11.1</b>	<b>Utvärdering av arbetet .....</b>	<b>61</b>
<b>11.2</b>	<b>Kritisk granskning av litteraturen .....</b>	<b>62</b>
<b>11.3</b>	<b>Erfarenheter under arbetets gång .....</b>	<b>62</b>
<b>11.4</b>	<b>Resultat.....</b>	<b>62</b>
<b>11.5</b>	<b>Förslag till fortsatt arbete .....</b>	<b>63</b>
	<b>BILAGA 1 - INTERVJUFRÅGOR .....</b>	<b>67</b>
	<b>BILAGA 2 – PRESENTATION AV FÖRFATTARE .....</b>	<b>68</b>

## 1 Introduktion

Detta examensarbete innehåller en studie av hur tekniker för participatory design kan användas tillsammans med de metoder och tekniker som används vid utveckling av informationssystem, i synnerhet inom traditionell systemutveckling.

En av de viktigaste egenskaperna hos ett datoriserat informationssystem är dess användbarhet. Om hänsyn inte tas till användarna vid systemutveckling blir risken stor för ett dåligt utformat system som inte är anpassat för sina användare och deras arbetsuppgifter. Detta leder till krånglig och ineffektiv användning och som följd försvinner ofta förväntade vinster av systemet då användarna gör fel, tvingas leta i manualer, fråga kollegor etc. Krångliga och svårförstådda system leder till missnöjda användare och minskad arbetsglädje (Rosengren och Wingstedt, 1993).

Syftet med ett informationssystem är att det skall förmedla information mellan människor i en organisation, och att vara ett stöd i arbetet, exempelvis i beslutsfattande (Andersen, 1994). Ett användbart informationssystem stödjer användarna med att genomföra arbetsuppgifterna. Ett användbart system inkluderar tillgänglighet av systemet genom interaktion via systemets gränssnitt, dess funktion och struktur, vilken stödjer användarnas aktiviteter i arbetet. Ett användbart system tillåter användarna att enkelt anpassa arbetet och expandera arbetskoncepten för att dra nytta av teknologin (Holtzblatt och Jones, 1993). Härav är det omöjligt att skapa ett ändamålsenligt informationssystem om man inte har klart för sig för vem eller vilken arbetsuppgift man skapar systemet (Andersen, 1994).

Att veta vilka som skall använda systemet blir ännu viktigare då organisationer är mycket dynamiska, det vill säga deras omgivning förändras mycket snabbt. Funktioner som planering, kontroll, organisering, och beslutsfattande måste kunna skötas effektivt om företaget skall kunna hålla sig kvar på marknaden. Detta gör att de måste kunna behandla data och använda information effektivt och på grund av detta blir informationen en kritisk faktor för att kunna genomföra det dagliga arbetet i företaget (Friedman, 1989). Källan är 12 år gammal men det går fortfarande att anta att organisationer är dynamiska, om inte ännu mer än i början av 1990-talet. Detta då företagen blir allt effektivare och informationen blir att viktigare.

Enligt Cherry och Macredie (1999) är de flesta industrier helt beroende av informationssystem och har lärt sig att utnyttja teknologin maximalt. I många företag är informationssystemen en hörnsten i organisationen. Sedan 1980-talet har man arbetat på att förbättra sättet att utveckla informationssystem och komma ifrån tänkandet om isolations- och dekompositionsproblem, det vill säga hur man skall bryta ner aktiviteter till atomära enheter och isolera dem som enskilda handlingar. Trots detta levereras fortfarande system som inte lever upp till utsatta krav och förväntningar. Vid utveckling av informationssystem används ofta traditionella systemutvecklingsmetoder. Dessa metoder använder användarrepresentanter i början av utvecklingsprocessen för att få fram en kravspecifikation, som sedan implementeras och testas på användarna. Tron på att datorapplikationer och systemutvecklingsmetoder tillsammans skall bota alla affärsproblem kan vara en av orsakerna till att många utvecklingsprojekt fortfarande inte lyckas att implementeras (Cherry och Macredie, 1999).

Ett viktigt faktum som ofta förbises i traditionella metoder är att det är användarna som är experter på sina områden. Utesluts användarna från systemutvecklings-

## 1 Introduktion

processen kan de känna sig hotade t.ex. genom rädsla för att förlora arbetet på grund av det nya informationssystemet. Etiketterna 'IT-användare' och 'IT-specialist' påvisar ännu mer skillnaden mellan de båda parterna, det vill säga att man placerar in de olika parterna i fack där den ena benämns specialist. Denna klyfta mellan systemutvecklare och användare gör att parterna ofta ser förbi varandra och missförstår varandra (Nurminen, 1988). Dessa problem kan snabbt leda till att systemutvecklingsprojektet blir till ett slagfält (Cherry och Macredie, 1999).

Om informationssystemet utvecklas utan involvering av användarna finns det stor risk att systemet kommer att avvisas eller inte användas av dem som det skall stödja. För att undvika detta måste användarna aktivt vara med i designen av det nya informationssystemet (Holtzblatt och Jones, 1993).

Av ovanstående kan man dra slutsatsen att det finns behov av utvecklare som vid utveckling av nya interaktiva informationssystem förstår eventuella användare och deras arbete. Frågan är hur utvecklarna skall få denna kunskap? Enligt Grudin (1993) är det idag många utvecklare som förlitar sig på sin intuition efter det att de lyssnat på och analyserat användarna för att få fram deras behov. Ett svar på frågan kan vara participatory design. Participatory design är ett svar på utvecklingsmetoder som ofta har svårt att tillfredsställa användarnas krav och med hjälp av denna teknik kan utveckling av informationssystem bli betydligt bättre (Grudin, 1993).

För att kunna utveckla bra interaktiva informationssystem krävs det att användarna involveras betydligt mer än vad de gör i traditionella systemutvecklingsmetoder, det vill säga metoder som sker stegvis och med tidig användarinblandning. Detta ställer hårda krav på de metoder och tekniker som används, men även på aktörerna i systemutvecklingsprojektet. Rapporten börjar med en bakgrund om traditionella systemutvecklingsmetoder, olika aspekter på dessa och participatory design. Diskussionen i ovanstående inledning används som underlag för problemformuleringen. Denna behandlar hur man kan använda olika tekniker inom participatory design i traditionella systemutvecklingsmetoder. Efter att problemområdet är definierat följer en fördjupad beskrivning av valda tekniker och hur dessa kan användas tillsammans med en traditionell systemutvecklingsmetod. Avslutningsvis redovisas resultatet följt av en diskussion runt detta.



## 2 Bakgrund

Ett system kan definieras som en organiserad samling människor, maskiner, procedurer, dokument, data eller andra entiteter som alla interagerar med varandra i en omgivning för att uppnå ett utsatt mål. Ett system är en term som används inom flera områden. Det gemensamma för system är att de har egenskaper som element, omgivning, interaktion mellan elementen och med omgivningen, men viktigast av allt så har systemet ett mål att uppnå (Ziya Aktas, 1987).

Ett informationssystem kan definieras som ett system som skall ge information inom en organisation, när informationen behövs och var den behövs för att leda företaget. Vidare är ett informationssystem en serie av affärsprocesser som är logiskt relaterade till varandra och hjälper till att uppnå organisationens mål (Ziya Aktas, 1987).

Företag som utvecklar system började växa fram på 1960- och 1970- talen. Deras princip var att sälja eller leasa hårdvara. Mjukvarufunktioner var sekundärt och människa-dator-gränssnitt fick mycket lite uppmärksamhet. Sedan dess har mjukvara tagit över hårdvarans plats och många nya produkter utvecklades under 1980-talet. Fram till sent 1980-tal låg fokus främst på funktionalitet och pris, det vill säga en produkt som uppfyller krav med avseende på tekniska aspekter till ett så lågt pris som möjligt. Idag så är funktionaliteten hos gränssnitten allt viktigare (Grudin, 1993).

Under de första decennierna inom systemutveckling var de flesta användarna ingenjörer och programmerare, vilket medförde att användarmedverkan inte var relevant då utvecklarna själva var representativa användare. Under de 15 senaste åren har detta ändrats, då datorer har blivit en viktig roll i alla företag (Grønbaek, Grudin, Bødker och Bannon, 1993).

Då metoder för systemutveckling togs fram lånades idéer från de metoder som användes i utveckling av mjukvara. Dessa metoder centrerar runt tre faser som täcker livscykeln för mjukvara och är huvudsakligen sekventiell. Processen, de tre faserna, kan delas upp i *definition* där problemet avgränsas, *utveckling* det vill säga konstruktionen av mjukvaran, och *underhåll* där eventuella justeringar sker. Varje del måste uppnå strikta kriterier innan nästa fas kan påbörjas. Varje fas måste också dokumenteras väl innan den avslutas (Cherry och Macredie, 1999).

Finns det formella fixerade krav är det inte svårt att konstruera mjukvara som möter dessa krav exakt. Mjukvarumetoder har visat sig vara mycket effektiva då dessa omständigheter gäller, det vill säga exakta och formella krav. Det är betydligt svårare att använda dessa metoder vid utveckling av informationssystem då de skiljer sig från mjukvaruutveckling genom att de ofta involverar manuella och sociala processer. En organisation innehåller mänskliga och sociala aktiviteter, vilket gör det omöjligt att bryta ner dessa till formella krav såsom vid mjukvaruutveckling. Härav är det svårt att följa de steg som metoden avser till punkt och pricka (Cherry och Macredie, 1999).

För att möta sociala aspekter har försök gjorts att producera metoder som integrerar de traditionella metoderna (se kapitel 2.4) med tekniker i människa-datorinteraktion (MDI). Dessa tekniker går ut på att använda användarnas kunskap för att skapa gränssnitt som underlättar inlärningsprocessen och förebygger negativ överföring, det vill säga ett gränssnitt som användarna inte förstår. Detta har dock enligt Cherry och Macredie (1999) misslyckat då kombinationen av traditionella dekomponeringsuppgifter och MDI-tekniker ofta resulterar i konflikter mellan användare och systemutvecklare, där användares önskemål ofta står i kontrast till den formella

## 2 Bakgrund

specifikationen. Detta kan leda till att användarnas önskemål används felaktigt i designen för att anpassas till specifikationen, vilket kan leda till ett användbart men oanvänt system (Cherry och Macredie, 1999).

I mitten av 1980-talet började forskare inom användarcentrerad design peka ut behovet av applikationer som inte bara är ”användarvänliga” utan även mer rotade bland dem som använder systemet, det vill säga att användarna förstår och accepterar systemet. Den användarcentrerade ansatsen försöker få in människorna i bilden och lyfta fram behovet av dessa människor i systemutvecklingsprocessen (Greenbaum, 1993).

Även om attityder håller på att förändras, så skapades de flesta affärsoperationerna och utvecklingsmetoder i de stora produktutvecklingsföretagen då hårdvara och mjukvarufunktionaliteten var det enda som var viktigt. Det är då inte överraskande att deras organisatoriska strukturer och processer inte stöder gränssnittsutveckling (Grudin, 1993).

Då människorna i organisationen är dynamiska och då organisationen i sig innehåller sociala och kulturella faktorer, tenderar dessa tidigare metoder att vara olämpliga vid utveckling av informationssystem. Designen av informationssystem är mycket komplex och kräver för att utvecklingen skall lyckas att integrationen mellan systemutvecklare och användare är mycket nära (Klößner, Pankoke-Babatz och Prinz, 1999). Yrkesaktiviteter i detta sociala system kan inte dekomponeras till atomära uppgifter utan att man förlorar omgivningen, kontexten, där uppgiften utförs (Cherry och Macredie, 1999). I stället måste systemutvecklarna få en grundlig förståelse av hela applikationsområdet och arbetsprocesserna för att få en lyckad designprocess (Klößner, m.fl., 1999). Är inte användarna med i utvecklingsprocessen är det risk för att utvecklarna vid analysen av dessa aktiviteter lägger in egna aspekter och erfarenheter i problemet. På grund av detta är det inte sällan som systemutveckling genom traditionell linjär dekomponering och återuppbyggnad misslyckas att implementeras då det är omöjligt att fånga dold kunskap, kunskap som användarna är omedvetna om (Klößner, m.fl., 1999).

### **2.1 Utvecklingsmetoder**

Då datorapplikationer börjades användas på 1960-talet var det få som implementerades med hjälp av en utvecklingsmetod. Detta då de flesta som använde dessa system var programmerarna själva. Då datorer blev allt vanligare i arbetslivet krävdes också bättre system av dem som skulle använda dessa. Detta ledde till tre huvudsakliga förändringar (Avison och Fitzgerald, 1998):

- Intresset för analys och design i utvecklingsprocessen ökade.
- Ökad komplexitet i organisationer ledde till önskan om mer integrerade informationssystem.
- Ökade önskemål om en accepterad metod för systemutveckling.

För att möta dessa förändringar började olika metoder att utvecklas. Definitionen på en metod är enligt Avison och Fitzgerald (1998) en samling procedurer, tekniker, verktyg och dokumentation, vilka hjälper systemutvecklare i deras arbete med att implementera ett nytt informationssystem.

Metoden består av olika faser och delfaser som leder systemutvecklaren genom arbetet. Olika tekniker som används hjälper systemutvecklaren att planera, leda,

kontrollera och utvärdera arbetet. En teknik är ett sätt att utföra aktiviteter i systemutvecklingsprocessen och lämpar sig olika bra i olika faser. I teknikerna används olika verktyg såsom papper och penna för att kunna utföra aktiviteten (Avison och Fitzgerald, 1998).

En metod består dock av mer än ovanstående begrepp. En metod bygger på något filosofiskt synsätt. Exempel på olika synsätt är mänskliga aspekter, vetenskapliga aspekter och pragmatiska aspekter (Avison och Fitzgerald, 1998).

### **2.2 Systemutvecklingens olika ansatser**

Det finns olika teman inom systemutveckling såsom (Avison och Fitzgerald, 1998):

- *Traditionell ansats*: är en modernare variant av den traditionella livscykeln, (se vidare i kap 2.4). Metoden använder tekniker som flödesdiagram och verktyg som CASE, exempel på traditionell metod är SSADM.
- *Objektorienterad ansats*: definierar objekt och attribut som tillsammans bygger upp informationssystemet. Fördelen med denna ansats är att systemet blir konsekvent då man använder objektorienterade egenskaper genom hela utvecklingsprocessen t.ex. objektorienterat programmeringsspråk vid realisering, objektorienterad analys och design.
- *Människo-orienterad ansats*: lägger fokus på användarna av systemet. Denna ansats försöker finna användarnas problem genom att ta hänsyn till sociotekniska aspekter och föreslå sätt att komma över dessa problem, exempel på metoder är ISAC och ETHICS,
- *Systemansats*: ser till informationssystemets helhet istället för att dela upp det i mindre delar, t.ex. SSM.
- *Öppen ansats*: innebär att man plockar aspekter från olika metoder och sätter ihop till en, det vill säga man använder delar av metoder där man tycker att de passar att användas, t.ex. multiview.

### **2.3 Skillnader mellan systemutvecklingsansatserna och den traditionella ansatsen**

Enligt Avison och Fitzgerald (1998) bygger systemansatsen på systemteori och försöker se systemets beteende som något stort och komplext. Ett system har indata, utdata och processer som konverterar indata till utdata. Principen är att betrakta en organisation som en helhet snarare än som fristående funktioner. För att få denna helhet är det viktigt att ta hänsyn till aspekter som förhållandet mellan organisationen och dess miljö, relationer mellan människa och teknik och respektives komplexitet. Utifrån detta tankesätt utvecklades metoden Soft System Methodology (SSM). I den traditionella ansatsen angrips det säkra och exakta i en specifik domän och man tittar på denna domän från en infallsvinkel. Detta är en stor skillnad från systemtänkandet där man istället försöker lägga sin kraft på de problem i den komplexa organisationen som inte är väldefinierade. Dessa odefinierbara problem kommer av att organisationen består av människor som är komplexa och oförutsägbara till sin natur (Avison och Fitzgerald, 1998).

## 2 Bakgrund

Den objektorienterade ansatsen skiljer sig från den traditionella då den bygger på helt andra koncept. I denna ansats ser man på informationssystemets olika komponenter som objekt t.ex. människan, aktivitet etc. Fördelen är att det går att använda sig av objektorientering genom hela utvecklingsprocessen, genom att använda objektanalys, objektmodellering, objektorienterade programmeringsspråk, etc. Det finns härav ingen anledning att översätta objekten till andra representationssätt och genom detta blir utvecklingen mycket konsistent. Detta är en stor kontrast till den traditionella ansatsen som använder olika representationer i olika processer. Objektorienteringen representerar allt som objekt; processer, människor etc. (Avison och Fitzgerald, 1998).

I den människo-orienterade ansatsen lägger man tyngdpunkten på informationssystemets användare och intressenter. Detta är ett stort kliv från den traditionella ansatsen där användarnas inblandning är frekvent betonad men där systemutvecklaren är den som håller i trådarna och fattar besluten. Den människo-orienterade ansatsen förespråkar att användarna skall vara med i utvecklingsprocessen för att informationssystemen skall bli lyckade. Detta då användarna känner att systemet blir relevant för deras arbeten om de får vara med under processen. Den största skillnaden mellan de båda ansatserna är att den traditionella fokuserar på teknologin: datorsystem, hårdvara och mjukvara. Utvecklingen drivs av teknisk expertis och användarna måste följa de regler och anvisningar som ges utan att avvika från dessa. I den människo-orienterade ansatsen ligger fokus på människorna i organisationen. Detta kan resultera i mindre system som inte är så komplexa och som inte är lika effektiva från en teknisk synvinkel, men som istället är lättare att hantera. Denna ansats förlitar sig mindre på teknologi och systemexperter och mer på användarnas kunskap (Avison och Fitzgerald, 1998).

Den människo-orienterade ansatsen kan ofta ses som oklar och svår att greppa. Metoderna är ofta mycket breda och innehåller många metodsteg. Detta gör att den lätt uppfattas som okontrollerbar, tidskrävande och svår att tillämpa. Den traditionella ansatsen ger mer kontroll över systemutvecklingsprocessen och är mer detaljerat beskriven, vilket gör att denna ansats används i större utsträckning än den människo-orienterade ansatsen. Dock finns det ett behov av att få in användarna mer aktivt i de traditionella ansatserna och av denna orsak kommer den traditionella ansatsen att behandlas i denna rapport. Som ett bra exempel på den traditionella ansatsen har metoden SSADM valts att studeras, eftersom den ligger mycket nära den traditionella livscykeln (se kap. 2.4), vilken kännetecknas av att vara väl detaljerad, regelmässig och ha låg involvering av användare. SSADM kan ses som ett alternativ till den traditionella livscykeln eller till och med som ett substitut till den (Avison och Fitzgerald, 1998). Det som i första hand gör dessa metoder lika är främst att de har en hög dokumentationsstandard, klara och detaljerade riktlinjer och kontinuerlig kvalitetsförsäkran (Avison och Fitzgerald, 1998). Vidare har det förekommit försök att integrera den traditionella metoden SSADM med den människo-orienterade metoden SSM. Resultatet av denna integration blev metoden COMPACT som kom 1989 (Avison och Fitzgerald, 1998). Då båda metoderna SSADM och COMPACT gjorts av samma företag skulle man kunna tro att COMPACT skulle ha hårdlanserats på samma sätt som SSADM. Detta lyckades med SSADM men då COMPACT är tämligen okänd har de tydligen inte lyckats lansera den med samma framgång. Av detta kan man dra slutsatsen att integrationen misslyckats.

### ***2.4 Den traditionella systemutvecklingens livscykel***

Den traditionella systemutvecklingsmetoden utvecklades runt 1970 och har sedan dess modifierats på olika sätt (Avison och Fitzgerald, 1998). Den traditionella systemutvecklingens livscykel är en metod med flera namn, exempelvis konventionell systemanalys, traditionell systemanalys och vattenfallsmodellen. De skiljer sig genom exempelvis med avseende på vilka delar i livscykeln som finns med, t.ex. förstudie och analys, arbetsfördelningen mellan användare och systemutvecklingsexperter, fokusering på process eller produkt (Avison och Fitzgerald, 1994).

Vid utveckling av ett informationssystem måste frågan om vad som skall uppnås ställas, innan diskussion om tekniska lösningar kan påbörjas. Stora delar av att utveckla ett informationssystem är att arbeta med beskrivningar. Det blivande informationssystemet växer fram genom att med olika beskrivningar först behandla överordnade frågor och till slut beskriva detaljerade tekniska förhållanden. Detta kräver olika sorters beskrivningstekniker och olika arbetssätt, t.ex. verksamhetsmodeller och grafiska beskrivningstekniker (Andersen, 1994).

Användarnas uppgift i systemutvecklingsprocessen är att redogöra för vad informationssystemet bör klara av för att underlätta deras arbete. Användarna behöver inte delta i, och skall i alla fall inte ha ansvaret för, val av tekniska lösningar. Det är experternas uppgift (Andersen, 1994).

Det som är karakteristiskt för den traditionella systemutvecklingens livscykel är att den består av sex olika delsteg, faser, i systemutvecklingsprocessen (Avison och Fitzgerald, 1998). Dessa är följande:

1. Förstudie
2. Systemkartläggning
3. Systemanalys
4. Systemdesign
5. Implementering
6. Test och underhåll

*Förstudien* består i att få förståelse för det nuvarande systemet och dess problem, samt att undersöka alternativa lösningar på dessa problem. För varje lösningsalternativ tar man hänsyn till tekniska, mänskliga och ekonomiska kostnader och förtjänst av att utveckla och operera systemet. Därefter rekommenderas en av lösningarna och lämnas till ledningen i en formell rapport.

Då ledningen accepterat förslaget på lösning inleds en *systemkartläggningsfas*. Detta innebär att samla in detaljerade fakta om systemet. Syftet är att utforska applikationsområdet, exempelvis funktionella krav i det nuvarande systemet och om dessa krav tillgodoses; begränsningar, datatyper och problem med nuvarande arbetsmetod. För att få fram denna information används exempelvis intervjuer med ledning och personal, enkäter, direkt observation av det intressanta området, och att analysera dokument relaterade till applikationsområdet.

Under processen är dokumentation en mycket viktig del, då utdata i en fas fungerar som indata i nästa fas. För att smidigt kunna dokumentera den information som tas fram används olika tekniker. Exempel på dessa är:

## 2 Bakgrund

- Flödesscheman som hjälper till att följa flödet av dokument genom en avdelning.
- Organisationsscheman vilket representerar den organisatoriska strukturen.
- Manuella dokumentspecifikationer som ger detaljer om dokument som används i manuella system.
- Rutscheman som visar hur olika komponenter i systemet interagerar med varandra.

Den tredje fasen är *systemanalys*. Här analyseras de fakta som samlats in om det nuvarande systemet. Frågor som ställs är: varför existerar problemet, varför infördes vissa arbetsmetoder, finns det alternativa arbetsmetoder, etc. Syftet är att få förståelse om alla aspekter i det gamla systemet och varför det ser ut som det gör. Genom detta går det att få en indikation på vad som skall förbättras i det nya systemet.

Nästa steg i utvecklingsarbetet är *systemdesign*. Denna fas skall leda fram till designen av det nya systemet. Systemdesignen innefattar både datoriserade och manuella delar av systemet. Liksom i tidigare steg dokumenteras resultatet noggrant, där följande detaljer skall finnas med:

- Indata och hur datan skall fångas (läggas in i systemet)
- Utdata från systemet
- Processer involverade i att konvertera input till output
- Strukturen av datoriserade och manuella filer
- Säkerhet och back-up
- Planer om systemtestning och implementation

Resultatet i denna fas är en detaljerad specifikation av systemet som sedan används vid implementeringen.

*Implementationen* innebär att det nya systemet realiseras, det vill säga specifikationen kodas för systemet. I denna fas testas systemet på användarna och dessa får även utbildning i det nya systemet. Ny hårdvara och mjukvara installeras.

Den slutliga fasen, *test och underhåll*, innebär att se till att systemet möter de krav som angivits och att utföra småförändringar på systemet då det är igång.

Andersens (1994) beskrivning på livscykeln är på ett ungefär densamma som Avison och Fitzgeralds (1998). Jämförs dessa två varianter av den traditionella livscykeln skiljer de sig inte nämnvärt. Båda metoderna börjar med en analys av systemet, både nuvarande och det framtida. För att samla in informationen utförs intervjuer, enkäter och observationer av användarna. Till dokumentation används tekniker som flödesscheman, organisationsscheman etc. Genomgående är också att användaren endast medverkar i de tidiga analysfaserna, genom intervjuer, enkäter och observationer, och att resterande del av systemutvecklingsprocessen styrs av systemutvecklarna. Båda beskrivningarna fokuserar på att samla och dokumentera information som sedan skrivs ner i en kravspecifikation, vilken sedan kravspecifikation används sedan som underlag för kodning av det nya systemet. Ändringar och modifieringar av det nya systemet görs först efter att det blivit implementerat och även detta sker med liten inblandning av användarna.

## 2 Bakgrund

Denna traditionella systemutvecklingsmetod har ett antal förutsättningar som talar för den (Avison och Fitzgerald, 1998). Den har blivit väl testad och prövad. Den grundliga dokumentationen försäkras en fullständig kravspecifikation och att denna dokumentation är kommunicerad via systemutvecklare, användare och operatörer. Metoden försäkras också att de som skall använda systemet får utbildning att kunna göra så. Slutligen är det även en fördel att metoden har delsteg, då en uppdelning gör att processen är lättare att kontrollera. Efter varje delsteg har systemutvecklare och användare tillfälle att utvärdera framstegen (Avison och Fitzgerald, 1998).

Några av dessa fördelar kan ifrågasättas. Utifrån litteraturen kan slutsatsen dras att det fortfarande är mycket vanligt att systemutvecklingsprojekt drar över tiden och överbudgeteras trots att de använder systemutvecklingsmetoder som har väl utsatta faser. Att arbeta efter förutbestämda faser kan snarare vara en nackdel än en fördel, eftersom verksamheterna är dynamiska och omständigheter hela tiden förändras. Detta leder till att tidigare krav måste modifieras, vilket ofta gör att det drar ut på tiden och, som följd av det, kostar mer pengar. Härav kan det vara svårt att följa stegen utan att gå tillbaka för att bemöta de förändringar som uppstår under arbetets gång. För att få med dessa förändringar i processen bör man ha ett iterativt tillvägagångssätt, det vill säga att man hela tiden går tillbaka i stegen och uppdaterar förändringar.

Vidare måste det nya systemet få acceptans av användarna innan de kan börjas utbildas i hur systemet fungerar. Får inte systemet denna acceptans har de redan förkastat systemet innan utbildningen börjat. Då användarna ofta inte får se det nya systemet förrän det är implementerat kan detta vara svårt att uppnå. Det kanske är bättre att lägga en del av resurserna för utbildning på att få med användarna i processen och på så sätt kanske lättare får användarna att acceptera det nya systemet.

Slutligen kan det ifrågasättas om det verkligen avsätts tid för systemutvecklare och användare att utvärdera framstegen efter varje fas, då användarna för det första inte är aktivt med i själva utvecklingsprocessen och för det andra då tidsbegränsningen ständigt gör sig påmind.

Avison och Fitzgerald (1998) visar på kritik som riktas mot den traditionella systemutvecklingen. Några exempel på kritik som riktats mot den är:

- *Misslyckande i att möta ledningens behov.* Detta har sitt upphov i att ledningen ofta ignoreras i utvecklingsprocessen, eftersom systemen utvecklades för rutinarbete.
- *Modeller i processen är ostabila.* Den traditionella metoden försöker förbättra processerna i organisationen, men processerna förändras frekvent i och med att omgivningen i organisationen förändras. Detta leder till att systemmodellprocessen ständigt måste skrivas om.
- *Utdatadriven design leder till dålig flexibilitet.* Utdata som skall produceras bestäms mycket tidigt i utvecklingsarbetet vilket gör att det kan vara mycket svårt att ändra i systemet om det sker förändringar i kravspecifikationen. Detta leder till att designen är oflexibel och är inte designen flexibel kan detta leda till att systemet är inaktuellt då det implementeras.
- *Otillfredsställda användare.* Då användarna inte kommer i kontakt med det nya systemet förrän det är implementerat är det inte förrän efteråt som de förstår vad deras olika krav får för effekter. Detta leder ofta till att systemet avvisas innan det har blivit helt färdigt.

Den första punkten kan diskuteras om den verkligen stämmer idag. I dag är organisationen mer uppbyggt som ett nätverk (Flensburg och Friis, 1999). Härav är ledningen lika beroende av informationssystemet som de på lägre befattningar. Idag används informationssystemet genom hela organisationen, inte bara för rutinarbete.

### **2.5 Participatory design**

Participatory design representerar ett tillvägagångssätt för att få med användarna aktivt i systemutvecklingsprocessen. Då det inte finns någon bra svensk översättning på detta begrepp kommer det engelska ordet, participatory design, att användas för att tillvägagångssättet inte skall uppfattas som något generellt utan som ett konkret begrepp. I fortsättningen kommer förkortningen PD att användas för participatory design.

PD representerar ett tillvägagångssätt att utveckla informationssystem som skall få in användaren i utvecklingsprocessen. Tillvägagångssättet är skapat i Skandinavien och går ut på att användarna av systemet spelar en kritisk roll i utvecklingsprocessen (Schuler och Namioka, 1993). Det finns fundamentala skillnader mellan PD och traditionell design (Schuler och Namioka, 1993, sid xi):

*"It rejects the assumption that the goal of computerization is to automate the skills of human workers, instead seeing it as an attempt to give workers better tools for doing their jobs.*

*It assumes that the workers themselves are in the best position to determine how to improve their work and their work life. In doing so, it turns the traditional designer-user relationship on its head, viewing the users as the experts-the ones with the most knowledge about what they do and what they need-and the designers as technical consultants.*

*It views the users' perception of technology as being at least as important as what they can do with it.*

*It views computers and computer-based applications not in isolation, but rather in the context of a workplace; as processes rather than as products."*

PD står i kontrast till specialistmodellen, där specialisten har befälet. PD kräver aktiv medverkan av användaren vid utvecklingsarbetet. Dock är PD inte emot expertis. Teknisk och personalspecialisering är viktig, men i PD får specialiseringen en annan roll. En relation mellan användare och experter måste skapas och båda parterna har ett ansvar för att utvecklingsprojektet skall lyckas (Schuler och Namioka, 1993).

Användarna utelämnas ofta från alla steg i den tekniska processen. De utelämnas från beslut angående vilken teknik som skall användas, hur utrustningen skall designas och hur arbetet skall designas. De är också utelämnade ur beslut om applikationerna i mjukvaran och systemet. Detta är inte bara odemokratiskt, det kan även få konsekvenser för arbetarnas hälsa, arbetstillfredsställelse och för arbetsprocessen i sig (Bravo, 1993).

Enligt Grudin (1993) är ett av målen med PD att förstå organisatoriska förändringar i datoranvändandet, effekterna av att introducera teknik i organisationsstrukturen och i processerna samt effekterna av organisatorisk återuppbyggnad angående sättet arbetet utförs på. Genom att reflektera över effekterna av nutida organisationsstrukturer och



## 2 Bakgrund

processer i produktutveckling och organisatoriska förändringar kan kanske hinder som skapats av nuvarande struktur och processer elimineras (Grudin, 1993).

PD är en inlärningsprocess i vilken designers och användare lär av varandra. PD kräver att deltagarna har förståelse om varandras bakgrund, kultur och språk. PD innebär inte bara att användarna medverkar i designen utan också att designers medverkar i användningen av systemet (Ehn, 1993).

Systemutvecklare i en användarstyrd utvecklingsmiljö måste spela en aktiv roll att fostra och möjliggöra för användarna att använda sina kunskaper i beslutsfattande. Detta arbete är en social aktivitet som involverar interaktionen mellan grupper av människor. Barriären mellan systemutvecklare och de som verkligen använder systemet måste brytas för att kunna bygga effektiv kommunikation under utvecklingsprocessen (Greenbaum, 1993). PD är ett sätt för systemutvecklare att få konkret feedback på systemutvecklingen under hela utvecklingsprocessen, som medför att systemet bättre kan passa in i den miljö det skall användas. PD kan hjälpa till att skapa en miljö där systemutvecklare och användare av datorapplikationer kan lära sig att utveckla system som bättre utför det arbete som skall göras (Greenbaum, 1993).

Vid PD ligger fokus på skillnader mellan olika mänskliga aktiviteter och datorutförande, genom att titta på vad människor gör med datorer, hur de använder datorer i samverkan med andra människor och vad de kanske kan göra bättre med datorer. I detta tillvägagångssätt involveras praktisk användning och förståelse av användarna, inte avsidig reflektion av användarna, som görs av systemutvecklaren efter t.ex. intervjuer och enkäter. Design betraktas som ett samspel mellan förståelse och skapande (Ehn, 1993).

Även om man använder PD är det inte säkert att projektet blir lyckat. En orsak till detta kan vara att de flesta systemutvecklingsprojekt fokuserar på att ta fram en produkt. För att kunna dra nytta av användarnas kunskap bör man istället fokusera på processen att utveckla produkten, eftersom organisationer består av olika sociala aktiviteter och processer och för att fånga dessa processer går det inte att endast fokusera på produkten. Fokuserar man bara på produkten utan att ta hänsyn till användarna går man miste om viktig information. För att få så hög kvalitet som möjligt på produkten måste användarna få vara med i processen. Läger man tyngden på processen leder detta till bättre kvalitet på produkten då användarna får känna sig mer ansvariga i utvecklingen (Grønbaek m.fl., 1993).

Designprocessens egenskaper beskrivs enligt Bødker m.fl. (1993) på följande sätt:

- En designprocess, som alla processer som äger rum i en organisation, är en politisk process, det vill säga demokrati på arbetsplatsen, och som ofta leder till konflikter då människor ofta har olika åsikter om saker. Dessa konflikter kan uppstå då olika parter i organisationen vill ha ut olika saker av det nya systemet. Ignorerar dessa konflikter kan lösningen bli mindre användbar och skapa ytterligare problem.
- Datorapplikationer som skapas för en arbetsplats behov kräver full användarmedverkan, vilket kräver träning.
- Designprocessen lyfter fram hur datorer används i organisationen.
- Genom att uppmuntra användarmedverkan får man upp ögonen för viktiga saker som annars skulle lämnats utanför den formella kravspecifikationen, så som dold kunskap och kommunikation.

## 2 Bakgrund

- För att komma åt användarnas dolda kunskap är det viktigt att simulera arbetssituationer.

Tar man inte hänsyn till dessa processer i sin omgivning kan detta hindra ett lyckat samarbete mellan systemutvecklare och användare och på så sätt misslyckas projektet.

Problem med PD är att det kan resultera i polarisering eller fragmentering av användargrupper och att det finns en möjlighet att manipulera valet av deltagare, så att de som anses vara "de rätta" får vara med i projektet. Detta kan få till följd att man inte får med de som verkligen bör vara med för att få önskad information (Avison och Fitzgerald, 1998). Ett annat problem är att systemutvecklarna kan motsätta sig att arbeta med användare då de anser att deras arbete tas över av någon annan (Avison och Fitzgerald, 1998).

Ännu ett problem är att när användarna skall vara med i utvecklingsprocessen blir alla beslut som fattas decentraliserade. Detta innebär att då användarna skall fatta beslut om förbättringar, förändringar, teknik och liknande finns det risk för att dessa centrala beslut leder till en lösning som verksamheten i helhet inte tjänar på (Andersen, 1994).

Det finns olika saker som kan hindra PD från att lyckas. Det kan exempelvis vara identifieringen och tidpunkten för inblandning av olika parter i projektet, olika sorters kontrakt, t.ex. ett kontrakt där priset är satt från början eller där det är flexibelt, och överenskommelser samt hur fokusering på process kan vara till större nytta än fokusering på produkt. För att överkomma dessa hinder i PD krävs att man finner tekniker som kan eliminera dessa hinder, att utveckla metoder och verktyg samt att förändra organisatorisk praxis och perspektiv. Kvaliteten i produkten och i processerna är den drivande kraften. För att utveckla bättre datorapplikationer, måste utvecklarna förstå processerna där det framtida systemet kommer att användas och på vilka systemet bygger. Traditionella sätt att tillhandahålla dessa processer, såsom användarmodeller och systembeskrivning, har misslyckats. Detta på grund av att de traditionella metoderna följer en viss ordning i metoden, vilket medför att det är svårt att hantera processerna när de förändras (Andersen, 1994). Genom att introducera PD-tekniker i de traditionella metoderna skulle det kanske gå att komma ifrån traditionella tekniker som användarmodeller och systembeskrivningar och kanske utveckla bättre system som är bättre anpassade till användarna.

### **2.6 Tekniker för *participatory design***

Många användare finner att det vid traditionellt utvecklingsarbete är tråkigt att delta i systemutvecklingsarbetet, därför är tekniker för PD ett bra alternativ då teknikerna ofta är lekfullt utvecklade (Bødker m.fl., 1991).

Det största hindret i PD är resurser. För att låta användarna vara med i utvecklingsprocessen krävs det att deras dagliga arbete minskas och att tid avsätts till systemutvecklingsprojektet. Får de inte denna tid, kommer de inte vara villiga att medverka i projektet under en längre period. Men resurser inkluderar inte bara tid och pengar utan även utbildning och hjälp av experter från olika områden. För att användarna inte skall dra sig ur projektet krävs det att designers argumenterar för olika resurser till användarnas fördel, exempelvis avsatt arbetstid för projektet (Bødker m.fl., 1991).

Designprojekt är oftast ett nytt område för de flesta användare. Teknikerna som beskrivs nedan har till uppgift att hjälpa till vid användarmedverkan i utvecklingsprocessen, då användare utan designerfarenhet kan ta lång tid på sig att lära sig,

## 2 Bakgrund

eftersom det nya området kan vara mycket svårt att förstå. För att detta skall kunna ske är det särskilt viktigt att användarna förstår sina möjligheter att förbättra organisationen då de medverkar. Vidare är det viktigt att de förstår vad som är viktigt och nödvändigt för deras egen situation på arbetsplatsen (Bødker m.fl., 1991).

I kapitel 2.6.1 till 2.6.5 beskrivs några resurser och tekniker som är nödvändiga för participatory design.

### 2.6.1 Framtidsseminarier

Framtidsseminarier har till uppgift att lyfta fram vanliga problemsituationer, generera en vision om framtiden och att diskutera hur dessa visioner skall realiseras. De som deltar i ett seminarium skall dela samma problematiska situation, och de skall vilja förändra denna situation (Kensinh och Madsen, 1991).

Ett framtidsseminarium leds vanligen av två seminarieledare, som ser till att alla får möjligheten att tala och att alla deltagare följer med i diskussionen. Detta kan ske genom att deltagarna skriver ned sina tankar på papper och sätter upp på väggen som sen kan diskuteras (Kensinh och Madsen, 1991).

Ett framtidsseminarium består av tre faser: kritisk fas, fantasifas och implementationsfas. Den kritiska fasen har till syfte att ta fram viktig information om organisationen. Ofta används strukturerad brainstorming för att få fram nuvarande problem i organisationen. Under fantasifasen tillåts de medverkande att komma med egna idéer och tankar om det framtida systemet. Inga förslag som kommer får här anses som för extrema. I den sista fasen, implementationsfasen, fokuserar man på resurser som krävs för att kunna göra realistiska förändringar. Dessa faser kompletteras kontinuerligt med uppföljning och förberedelser även detta i seminarieform (Bødker m.fl., 1993).

Ett framtidsseminarium hjälper till att definiera systemets mål och fokusera på aspekter som har med det egentliga arbetet att göra istället för på tekniska och ekonomiska aspekter. Tekniken gör att användaren kan ta en aktiv roll i utvecklingen och delge sin kunskap till projektet och därmed få fram de verkliga problemen (Kensinh och Madsen, 1991).

Framtidsseminarier är en bra teknik att använda då medarbetarna är oerfarna i systemutveckling. I dessa seminarier kan både användare och systemutvecklare lära av varandra. Användarna kan lära sig om teknisk design medan systemutvecklarna lär sig om organisationen och olika attityder i arbetet. Framtidsseminarier fungerar som ett forum där användare kan få den kunskap som de anser sig behöva (Cherry och Macredie, 1999).

Vidare används ofta brainstorming i arbetsgruppen. Detta ger möjligheten att få nya sätt att arbeta på, baserat på kollektiv erfarenhet av de som skall använda systemet. Skulle inte användarna vara med finns det en risk att systemanalytiker och ledning skulle ha svårt att hitta dessa arbetssätt (Cherry och Macredie, 1999).

Det finns vissa praktiska problem med framtidsseminarier. Exempel på detta är tidspress under seminarierna, försäkring om att planerna verkligen utförs, välja ut vilka deltagare som skall medverka och försäkran om att seminarieledarna inspirerar utan att påverka (Kensing och Madsen, 1991).

Ett annat problem med framtidsseminarier är att blanda olika grupper med människor. Finns exempelvis ledning och arbetare med under samma seminarium kan arbetarna

vara rädda för att t.ex. kritisera organisationen, vilket gör att syftet med seminariet går förlorat (Bødker m.fl., 1991). Likaså är det vanligt att ledningen har lättare att uttrycka sig än arbetare, vilket gör att det är lätt att ledningen tar kontroll över seminariet (Bødker m.fl., 1993).

### 2.6.2 Design genom rollspel

Denna teknik går ut på att man skall hitta olika alternativa arbetsorganisationer genom att ”spela” dem. På detta sätt får de medverkande konfronteras med de olika problem som finns i organisationen genom att spela teater om dem. Idén bakom denna teknik är att:

- Den gör en skillnad för de medverkande
- Implementationen av resultatet är sannolikt
- De är roliga att delta i

De två första punkterna involverar den politiska sidan i PD; användarna måste garanteras att deras designarbete tas på allvar. Den sista punkten involverar själva designprocessen (Ehn och Sjögren, 1991).

I denna teknik betraktas designarbetet som en teaterpjäs där designarbetet delas upp i akter. Ehn och Sjögren (1991) bedrev ett projekt för Konsumentverket där utvecklingsarbetet bestod i en prolog och tre akter. Idén med detta spel är att det har likheter med vanliga spel, exempelvis monopol, med en spelplan, pjäser etc. Innan spelet kan börja gör man i ordning scenen. Här presenteras designernas syfte med spelet och grundläggande regler. Vidare introduceras även arbetsplatsen för systemutvecklarna. En spelplan i form av en stor tavla gjordes i ordning där man kunde sätta upp olika villkor och förpliktelser. Tavlan var uppdelad i tolv delar som var och en representerade ett arbetsområde. Under prologen sattes även rollistan, som var uppdelad efter deltagarnas professionella ambitioner och organisatoriska önskemål. Deltagarna förberedde sig inför akt 1 genom att studera ett manuskript och att skriva situationskort som beskrev existerande problem (Ehn och Sjögren, 1991).

I första akten började själva spelet. Deltagarna var placerade på scenen framför spelplanen där en aktör om gången tog ett situationskort, läste upp det och den som hade den roll som det aktuella problemet refererade till skulle då komma med ett utlåtande om problemet, exempelvis vem tar på sig ansvaret att byta färgpatron i skrivaren? Den som tog på sig detta ansvar gjorde förpliktelsen under vissa villkor såsom ”Jag beställer nya patroner om jag får uppskattning för att jag gör det”. Problemet diskuterades av hela gruppen innan nästa aktör tog ett nytt kort. På detta sätt skapades simuleringar av problemsituationer. Den första akten avslutades med en diskussion och förberedelser inför akt två.

I den andra akten byggdes en spelplats upp där det verkliga arbetet simulerades. Här låg fokuseringen på de situationer där systemet kraschade. Detta visade sig mer komplicerat än akt ett och därav infördes systemutvecklingskort som skulle visa olika lösningar på de olika problemen. Lösningarna på problemen kunde antingen antas eller förpassas. Innan akt tre började hade en mängd lösningar antagits och dessa uppdaterades på spelplanen.

I den sista akten överförde man spelet till verklighet. Detta bestod i att överföra resultatet på spelplanen till en aktionsplan som bestod av de krav man fått fram genom

## 2 Bakgrund

spelet. Dessa krav utvärderades sedan och ledde fram till en kravspekifikation (Ehn och Sjögren, 1991).

Fördelen med denna sorts spel är att de stöder och uppmuntrar inblandning i utvecklingsprocessen, inte bara för att det är roligt, utan för att tekniken tar användarna på allvar (Bødker m.fl., 1993).

Dock är nackdelen att det är ett mycket tidskrävande sätt att gå till väga, detta leder i sin tur att det blir dyrt att använda denna teknik. Men i och med att erfarenheten om spelen ökar så minskar också tid och kostnad (Ehn och Sjögren, 1991).

Kritik som riktats mot denna teknik är att man gått från seriös demokratisering i organisationer till ett ytligt sällskapsspel (Ehn och Sjögren, 1991). Denna kritik är inte helt obefogad. Är det verkligen så lätt att ta detta tillvägagångssätt på allvar. Kan det inte vara svårt att få 'Oskar, montör, 57 år' att delta i ett rollspel om systemutveckling, där han ska spela rollen som 'Mia, sekreterare, 25 år'?

Om denna teknik skall vara möjlig bör den nog tillämpas i mindre organisationer där medarbetare har en närmare relation än i stora organisationer. Detta på grund av att det kan vara svårt att få deltagarna så avslappnade att de verkligen vågar uttrycka sig genom rollspel.

### 2.6.3 Mock-up design

Då ingen översättning på 'mock-up' har hittats används det engelska ordet.

'Mock-up' innebär att man simulerar verkligheten med enkla medel. Man simulerar de problem som finns, framtiden och implementationen genom att använda modeller av exempelvis mjukvara utan att bygga in funktionalitet (Ehn och Kyng, 1991). Den används för att utvärdera designen, för att få idéer till modifiering eller nya radikala designar och skall fungera som ett medium för samarbete för att nå förändringar (Ehn och Kyng, 1991). Denna teknik skall hjälpa användare så väl som utvecklare att minska gapet mellan verklighet och fantasi för att få fram lösningar på olika problem. I motsats till rena beskrivningar, så påminner 'mock-ups' användarna om de dagliga arbets-situationerna. Några av orsakerna till varför 'mock-ups' fungerar är:

- Tekniken uppmuntrar användarna att utnyttja sina egna erfarenheter
- Tekniken är lätt att förstå då det inte går att förväxla verklighet med fantasi och alla kan vara med och modifiera dessa begrepp
- Tekniken är ej dyr att använda
- Tekniken är rolig att arbeta med

(Bødker m.fl., 1993).

'Mock-up' blir användbar då den verkar vettig för de som medverkar eftersom man använder ett språk som gör att de förstår var som menas, inte för att den speglar en verklig händelse utan att de förstår interaktionen och relationen mellan olika händelser (Ehn och Kyng, 1991).

Fördelarna med 'Mock-up' är att tekniken kan användas med hjälp av billigt material, det vill säga man behöver inte använda datoriserade hjälpmedel utan kan klara sig med papper och penna. Vidare så är detta enkla material känt för de flesta, vilket gör att alla känner sig hemma i materialet och det är ingen förväxling mellan verklighet och

simulering. En annan fördel är att det är mycket enkelt att modifiera systemet då det görs med så enkla medel. Genom att använda en sax går det lätt att ta bort sådant som är fel och lika enkelt att lägga till nya funktioner. Likaså är det lätt för användarna att förstå modifieringarna (Bødker m.fl., 1993).

'Mock-up' har dock även begränsningar. Att göra förändringar i 'Mock-up' kan vara mycket tidskrävande. Om man exempelvis har gjort en design av menyer genom 'mock-up' och måste förändra dessa kan det kräva ett stort antal ritningar för att ändra alla menyer. En annan nackdel är att det är svårt att gestalta beteende hos applikationer med 'mock-up'. Det är svårt att få det realistiskt (Bødker m.fl., 1993).

### 2.6.4 Kooperativ prototyping

Kooperativ prototyping skiljer sig från traditionell prototyping, där utvecklarens perspektiv står i fokus. Det vill säga, designers utför analyser i organisationen och utvecklar egna prototyper som sedan testas på användarna och sedan tar emot feedback om systemet. Traditionell prototyping tar lite hänsyn till själva designprocessen medan kooperativ prototyping sätter användarnas lärande om tekniken i fokus (Bødker m.fl., 1993).

Kooperativ prototyping riktar sig åt att etablera en designprocess där både användare och designers medverkar aktivt och kreativt genom att utnyttja sin egen kompetens. För att stödja denna process måste designern på något sätt låta användarna uppleva den nuvarande arbetssituationen med den framtida applikationen; det vill säga användarnas nuvarande kunskap måste komma i kontakt med nya tekniska möjligheter (Bødker och Grønbæk, 1991). Genom att låta användarna använda systemet medan det utvecklas kan man fånga de situationer som är problematiska och tillsammans kan användare och utvecklare analysera problemet då det uppstår och modifieras (Bødker m.fl., 1993).

Prototyping tillämpas med fördel i analysfasen av ett systemutvecklingsprojekt där den kan komplettera de abstrakta beskrivningar av verksamheten i form av t.ex. informations- och affärsprocessbeskrivningar som ofta tas fram där. En prototyp kan bidra till ökad förståelse för de praktiska konsekvenserna av modellerna för användarnas arbete. Detta gäller speciellt om analysen av verksamheten har resulterat i att den skall förändras, t.ex. att arbetet skall genomföras på ett nytt och mer effektivt sätt. Det kan då vara svårt att föreställa sig konsekvenserna för det dagliga arbetet. På så vis kan prototypen dels bidra till att stämma av specifikationerna och förankra dessa hos användarna, dels ge rekommendationer till det framtida utvecklingsprojektet. Tekniken kan dock även användas för prototyputförande av det slutgiltiga gränssnittet (Rosengren och Wingstedt, 1993).

Ytterligare fördelar är att de blivande användarna genast kan ge sina synpunkter och systemutvecklarna kan då snabbt korrigera felen (Flensburg och Friis, 1999). Nackdelen är att prototypen endast kan ses som en kravspecifikation och det kan ta lång tid innan det färdiga systemet kan tas i drift. Risker är då stora att organisationen hinner förändras innan det nya systemet tas i drift och blir då inaktuellt innan det är färdigt. En annan risk är att användarna blir otåliga i väntan på det nya systemet, vilket kan leda till konflikter. Dock finns det sätt att lösa dessa problem. Ett sätt är att tillfälligt ta prototypen i drift (Flensburg och Friis, 1999), detta är även en fördel då

problem med prototypen uppstår ser man felet direkt och kan rätta till det (Cherry och Macredie, 1999).

Vid kooperativ prototyping fokuserar man mest på innehållet och funktionen hos systemet. Eftersom det är en iterativ process kan det vara svårt att fastställa dels hur lång tid det tar att utveckla det nya systemet och dels hur mycket det kommer att kosta. Detta leder till att företag kan vara emot kooperativ prototyping i för stor utsträckning. Dock vet man att kooperativ prototyping ger snabbare delresultat i utvecklingsprocessen och att användare får ett mer gemensamt språk med data-specialister och systemutvecklare (Flensburg och Friis, 1999).

### 2.6.5 Kontextuell utfrågning

Kontextuell utfrågning är en teknik som hjälper användarna att delta i design av informationssystem. Tekniken stöder arbetet mellan systemutvecklare och användare för att lättare få fram informationen som behövs från användaren. Tekniken bygger på tre koncept som guidar genom systemutvecklingsprocessen. Dessa är:

- Kontext
- Kompanjonskap
- Fokus

Det första konceptet bygger på att det bästa sättet att förstå människors arbete är att prata med dem i deras arbetsmiljö. I informella intervjuer med användare är det lätt att de generaliserar sitt arbete och talar om hur det är tänkt att det skall gå till, men det är sällan det utförs som det står på pappret att det skall utföras. Informationen om det verkliga arbetssättet är lättast att få fram då man pratar om det där arbetet utförs. På detta sätt får man information om problem då de verkligen uppstår och ser direkt vad som kan göras för att lösa dessa problem. Kontextuell utfrågning fokuserar på att låta människor berätta om sitt arbete, hur de gör, varför de gör på ett visst sätt, etc. för att få fram riktig information (Holtzblatt och Jones, 1993).

Det andra konceptet, kompanjonskap, kommer från behovet att förstå användarnas erfarenheter av arbetet och systemet. Denna information är oftast osynlig och kan inte nås genom att stå utanför arbetsprocessen. Det måste finnas en dialog mellan systemutvecklare och användare som bygger på ett kompanjonskap. På detta sätt kan de båda parterna tillsammans få förståelse om arbetet och från detta utröna tekniska möjligheter och problem som uppkommer i arbetet. Utifrån denna förståelse utvecklar sedan designers systemet (Holtzblatt och Jones, 1993).

Den sista punkten, fokus, innebär att systemutvecklarna filtrerar den information de får och fokuserar på en sak åt gången. Genom detta erhålls den information som är viktig och den information som är mindre viktig ignoreras. Fokus expanderas sedan allt eftersom mer information samlas in (Holtzblatt och Jones, 1993).

Fördelen med kontextuell utfrågning är att den kan användas i systemutvecklingscykeln för att grundlägga systemutvecklarnas förståelse för arbetsaktiviteterna. Likaså är den bra att tillämpa på andra tekniker såsom prototyping och seminarier.

Kontextuell utfrågning kan användas genom hela utvecklingsprocessen. Ofta har den tillämpats tillsammans med vattenfallsmodellen (Holtzblatt och Jones, 1993). Vidare är kontextuell utfrågning mer ett tankesätt än en teknik. Detta då exempelvis prototyping

## 2 Bakgrund

används tillsammans med kontextuell utfrågning, som ett tankesätt. Det vill säga man tänker i termer som kontext, fokus och kompanjonskap. Egentligen skiljer den sig inte så mycket från ”vanliga” tillvägagångssätt såsom intervjuer och observationer. Dock är det positivt att man kan använda detta tillvägagångssätt tillsammans med en mer traditionell metod. Nackdelen kan vara att det tar mycket lång tid att samla in information om man skall besöka varje arbetsstation i en organisation för att följa med under tiden de arbetar och se vad som är och inte är ett problem.

Likheterna mellan dessa olika tekniker är att användarna står i fokus, medan systemutvecklarnas syfte är att backa upp och stödja användarna i diskussionerna. Vidare verkar teknikerna till största delen användas i analys- och designfaserna, det vill säga de faser som leder fram till en kravspecifikation. Den enda tekniken som skulle kunna användas efter kravspecifikationen är kooperativ prototyping där användarna kan vara med under realiseringen. Vidare är det egentligen ingen teknik som används helt enskilt, det är ofta en kombination av alla tekniker som tillämpas. I kooperativ prototyping använder man sig kanske av framtidsseminarier för att finna problem som sedan löses med hjälp av prototyping.



### 3 Problemställning

Utifrån litteraturen går det att utröna att det idag är många informationssystem som aldrig blir implementerade på grund av att de inte når upp till de krav som användarna har på systemet. Informationssystemet får ingen acceptans av användarna och blir på så sätt oanvändbart. Orsaken till detta är inte tekniska utan har snarare med människan att göra. För att systemutvecklarna skall kunna designa funktionella och kravenliga informationssystem krävs det att användarna får vara med i designprocessen och påverka systemets utseende och funktionalitet. Det är användarna som är experter på sitt område, inte systemutvecklarna.

Vidare säger även litteraturen att för att användarna skall kunna medverka aktivt i systemutvecklingsprocessen måste systemutvecklingsprocessen förändras från att ha varit, och är, fokuserad på teknik, produkt och utvecklingsexperter, till att fokusera på de som skall använda systemet, deras processer och deras kunskap. Detta innebär inte bara att ändra på metoder utan även på tekniker och verktyg som används i processen. Idag används ofta metoder, såsom traditionella metoder, som bygger på att utvecklingen sker sekventiellt. På grund av tidigare nämnda problem med de traditionella metoderna måste systemutvecklarna lära sig nya tillvägagångssätt att få fram den kunskap som behövs för att bygga bra informationssystem. Det behövs helt enkelt ett nytänkande inom systemutvecklingen.

Detta arbete handlar om att integrera dessa traditionella metoder, i detta fall SSADM, med participatory design för att på ett bättre sätt få fram information från användarna. Genom att använda de tekniker som finns i PD tillsammans med en traditionell systemutvecklingsmetod får man dels med användarna, som är experter på sitt område, och dels behålla kontrollen över systemutvecklingen med hjälp av utsatta steg och riktlinjer enligt metoden. Detta kan leda till bättre system då användarna aktivt är med i utvecklingsprocessen, samtidigt som det är lättare att hålla tidsbegränsningar och budget då man ändå följer en specifik metod. Detta kan i sin tur ge en ökad acceptans bland användarna och på så sätt öka chanserna för ett lyckat systemutvecklingsprojekt. Resultatet blir ett informationssystem som är både användbart och välanvänt.

#### 3.1 Examensarbetets frågeställningar

Detta examensarbete kommer att innehålla en utredning om tekniker som används i participatory design och hur dessa kan användas tillsammans med traditionella systemutvecklingsmetoder. Detta leder till huvudfrågan i arbetet:

- Går det att integrera den traditionella ansatsen med participatory design?

För att nå fram till huvudmålet kommer följande delmål att behandlas:

- Hur används användarmedverkan i systemutvecklingsmetoden SSADM?
  - När?
  - Hur ofta?
  - På vilket sätt?
  - Med vilket syfte?

I detta delmål vill jag undersöka när i systemutvecklingsprocessen användarmedverkan förekommer, och hur ofta den förekommer. Vidare vill jag titta på

### 3 Problemställning

hur man går till väga för att få informationen av användarna, det vill säga på vilket sätt och med vilket syfte. Syftet med detta delmål är att få ett underlag för att använda i undersökningen av huvudmålet.

- Kan PD-teknikerna framtidsseminarium och kooperativ prototyping användas i SSADM?
  - När?
  - Hur ofta?
  - På vilket sätt?
  - Med vilket syfte?

Här vill jag titta på när i processen det är lämpligt att använda teknikerna och hur ofta detta skall göras. Jag vill också utreda på vilket sätt teknikerna skall användas tillsammans med en traditionell metod som SSADM.

#### **3.2 Avgränsning**

I detta arbete kommer jag inte att behandla alla de tekniker som finns inom participatory design utan kommer att begränsa mig till två stycken. Jag har valt framtidsseminarier och kooperativ prototyping då jag anser att dessa verkar mest lämpade och trovärdiga att använda tillsammans med en traditionell systemutvecklingsmetod då de är relativt flexibla och verkar mest konkreta att använda t.ex. i analysfasen. De övriga teknikerna verkar vara oklara och svåra att greppa, det vill säga det är svårt att förstå dem och hur de skall användas. Detta kan vara en nackdel då de skall integreras med en traditionell metod.

Jag kommer inte heller att göra en generell översikt över alla traditionella metoder utan har valt ut en, SSADM. Detta för de traditionella metoderna är likartade och för att istället kunna gå djupare in på ämnet.

#### **3.3 Förväntade resultat**

Det resultat jag förväntar mig att komma fram till är att det är möjligt att integrera PD-tekniker med en traditionell metod som SSADM. Vidare vill jag undersöka hur detta skall kunna göras på ett bra sätt och när man skall använda dessa tekniker i utvecklingsprocessen. Jag tror att det är möjligt att använda sig av PD-tekniker i analys- och designfaserna för att få fram en kravspecifikation men däremot tror jag inte att det går att kombinera dessa tekniker med SSADM i implementationsfasen. Problem som kan tänkas uppstå är att de olika deltagarna kan ha olika mål och önskningar angående det nya informationssystemet, vilket kan resultera i svårlösta konflikter. Ett annat problem är att det kan vara svårt att hålla sig till de olika faserna i SSADM vid inblandning av PD-tekniker då dessa är mer ostrukturerade och kan dra ut på tiden.

## 4 Metod

Detta kapitel behandlar olika tillvägagångssätt för att undersöka frågeställningen i examensarbetet. Kapitlet inleds med presentation och diskussion av de olika metoder som är möjliga att använda för att besvara frågan. Även de fördelar och nackdelar som följer med de olika metoderna tas upp. Utefter denna diskussion motiveras vald metod och kapitlet avslutas med en plan för hur arbetet med frågeställningen skall läggas upp.

### 4.1 Möjliga metoder

De metoder som är möjliga att använda i detta arbete är:

- Intervjuer
- Fallstudie
- Litteraturstudie

varav litteraturstudie och intervjuer har valts.

#### 4.1.1 Intervjuer

Intervjuer är en teknik för att samla in information som bygger på att intervjuaren kontakter intervjupersonen och ställer frågor (Patel och Davidson, 1994). Fördelen med intervjuer i jämförelse med enkäter är att många och krångliga frågor kan ställas, det är enkelt att reda ut oklarheter i frågorna och öppna frågor kan lättare besvaras (Dahlström, 1996). Vidare ger intervjuer ofta utförliga svar med hög kvalitet. Då arbetets frågeställning kräver en kvalitativ undersökning så lämpar det sig att använda intervjuer. Detta då en kvalitativ undersökning bygger på att få förståelse för ett problemområde i sin helhet, inte enskilda delar av problemområdet (Patel och Davidson, 1994). Den största nackdelen med intervjuer är att de är mycket tidskrävande. Ytterligare en nackdel är svårigheten att hitta rätt personer att intervjua (Dahlström, 1996).

För att besvara frågeställningen kan intervjuer vara bra för att undersöka om systemutvecklare med erfarenhet av traditionell systemutveckling idag tror att det är bra att kombinera traditionella metoder med PD. Vidare kan man få svar på när PD-teknikerna skulle kunna vara lämpliga att använda i utvecklingsprocessen och om det är möjligt att använda dem. I detta arbete kommer intervjuer användas som ett stöd till litteraturstudien för att dels få en bild av hur användarna deltar i systemutvecklingsprojekt där traditionella metoder används i verkligheten och dels hur systemutvecklare ser på användarmedverkan och PD.

#### 4.1.2 Fallstudie

Fallstudie är en metod som innebär att vi gör en djupgående undersökning av en situation, en individ, en grupp eller en organisation (Dawson, 2000). Fallstudier används ofta för att studera förändringar och processer och utgår från ett helhetsperspektiv för att få så täckande information som möjligt (Patel och Davidson, 1994). Det är ibland också nödvändigt att studera fler än ett fall, exempelvis två olika projekt (Patel och Davidson, 1994). Fallstudie skulle kunna användas i arbetet för att besvara

frågeställningen. Här skulle det då gå att titta på olika systemutvecklingsprojekt som använder olika traditionella utvecklingsmetoder och se om det går att använda PD tillsammans med dessa.

För att få ett intressant resultat i arbetet skulle det vara bra att kunna pröva om det går att integrera en traditionell systemutvecklingsmetod med PD-tekniker i verkligheten. Vidare skulle det också genom en fallstudie vara möjligt att se varför det inte skulle gå att integrera dessa två ansatser om så var fallet. Nackdelen med denna metod är att det är svårt att hitta fall att studera. För att besvara frågeställningen skulle det behövas något eller några systemutvecklingsprojekt som använder sig av olika traditionella metoder och dessutom var villiga att prova PD-tekniker. Fallstudier är också mycket tidskrävande och på grund av dessa orsaker är det näst intill omöjligt att utföra en fallstudie i detta arbete.

### 4.1.3 Litteraturstudie

De vanligaste källorna där vi hämtar kunskap är böcker, rapporter och artiklar som är publicerade i vetenskapliga tidskrifter. I böcker finns oftast försök att sammanställa och systematisera den kunskap som finns inom ett problemområde och det är här det går att finna teorier och modeller utvecklade i sin helhet. I artiklar, rapporter och konferensskrifter finns de absolut senaste rönen (Patel och Davidson, 1994).

Ett av de viktigaste inslagen i en litteraturstudie är att den litteratur man använder är trovärdig. För att kunna göra denna bedömning är det viktigt att granska litteraturen kritiskt. Detta innebär att vi måste ta reda på när och var litteraturen tillkommit, varför den tillkommit och vad dess syfte är. Slutligen är det också mycket viktigt att granska författaren (Patel och Davidson, 1994). För att litteraturen skall ses som sanningsenlig bör den vara känd. Detta innebär att den skall vara refererad i relaterad litteratur. Om många andra författare refererat till ett visst verk innebär detta ofta att verket är av betydelse inom området och kan därav antas vara sanningsenligt (Dawson, 2000).

Fördelen med att använda litteraturstudie för att besvara problemställningen i detta arbete är att det finns mycket litteratur inom området PD och traditionella metoder. Mycket av det material som finns bygger på fältstudier där man använt PD i systemutvecklingsprojekt och har därmed dokumenterade resultat. Härav finns det mycket praktisk erfarenhet inom området att använda sig av.

Nackdelen med litteraturstudier är att det är mycket tidskrävande att sätta sig in i materialet. Det kan vara svårt att sortera ut det som är väsentligt då olika tidigare studier fokuserar på olika aspekter av ämnet. Härav är det åter viktigt att granska källan så att man får klart för sig vilket syftet med studien är.

## 4.2 Val av metod

Då arbetet är en kvalitativ undersökning behövs underlag som är utförligt och väl beskrivet. De tre ovan beskrivna metoderna skulle alla kunna bidra till denna information. Det är dock svårt att genomföra en fallstudie då den begränsade tiden gör att det är svårt att hitta lämpliga fall att undersöka. Idag finns det relativt mycket material inom området PD. Det finns studier där man använt PD inom systemutvecklingsprojekt och även dokumenterade fördelar och nackdelar med PD, däremot inte om PD tillsammans med traditionell utveckling. Intervjuer skulle vara bra för att

undersöka om PD-teknikerna skulle gå att använda i integration med traditionella metoder i allmänhet.

De metoder jag använder mig av för att besvara min frågeställning är en litteraturstudie med stöd från intervjuer. Med hjälp av litteraturen kommer de två delmålen undersökas för att sedan tillsammans med intervjuer användas för att besvara huvudmålet i arbetet. Vid en litteraturstudie inhämtas kunskap om de båda delmålen genom att studera litteratur som behandlar systemutvecklingsmetoden SSADM och teknikerna framtidsseminarium och kooperativ prototyping. Jag börjar med en kort presentation av metoden som helhet för att sedan lägga fokus på användarmedverkan i SSADM. Genom att intervjua personer som arbetar med systemutveckling skall jag sedan undersöka om de tror att det går att integrera PD-tekniker med traditionell systemutveckling. Resultatet från de båda delmålen och intervjuerna kommer sedan att ligga till grund för huvudmålet.

### **4.3 Plan för arbetet**

Det första delmålet innebär att göra en utförlig beskrivning av systemutvecklingsmetoden SSADM och PD-teknikerna framtidsseminarium och kooperativ prototyping. I detta kapitel behandlas metodsteg, tekniker och användarmedverkan i SSADM. Metoden kommer att beskrivas stegvis, det vill säga beskriva vad som görs i varje steg och vilka tekniker som används.

Det andra delmålet innebär att undersöka om PD-teknikerna går att integrera med SSADM. Detta görs genom att fokusera på användarmedverkan i metodstegen, när det används, hur ofta och om det är möjligt att använda PD-teknikerna i respektive steg. Integrationen innebär att se om det är möjligt att byta ut några av SSADMs tekniker mot PD-tekniker eller om det går att lägga till PD-tekniker till de som finns i SSADM. Analysen omfattar att varje metodsteg beskrivs separat för att sedan undersöka om det går att integrera respektive metodsteg med PD-tekniker.

Efter litteraturstudien kommer intervjuer att utföras på ett fåtal personer för att ta reda på huruvida det går att integrera PD-tekniker med traditionella systemutvecklingsmetoder. För att genomföra intervjuerna har jag vänt mig till personer som idag arbetar med systemutveckling och använder någon systemutvecklingsmetod. Storlek och typ av företag har ingen betydelse då samarbete med användarna existerar i systemutvecklingsprojekt oavsett företag. De egenskaper som intervjupersonerna skall ha är att de skall ha erfarenhet av systemutvecklingsprojekt och vara vana att arbeta med metoder. För att få den information som är nödvändig skall tre till fyra personer intervjuas men går det inte att få tag i det antalet används ändå intervjuerna som stöd till litteraturstudien för att besvara huvudfrågan.

Intervjuerna skall vara direktintervjuer för att kunna hålla diskussionen så öppen som möjligt. Telefonintervjuer kommer inte att användas då frågorna är formulerade för att kunna diskuteras, vilket kan vara svårt via telefon då det är lätt att missa viktig information eftersom samtalet inte kan bandas. Intervjupersonerna skall arbeta på orter inom närliggande område, detta då det dels tar lång tid att ordna en intervju på annan ort och dels för att slippa resa. Vid direktintervjuerna kommer samtalet att spelas in på band för att inte missa någon information.

För att besvara arbetets frågeställning med hjälp av intervjuer bör frågorna vara ostrukturerade frågor med låg standardisering, detta då frågeställningen kräver att

frågorna är öppna så att intervjuaren och intervjupersonen fritt kan diskutera frågorna och följdfrågor under tiden de ställs (Patel och Davidson, 1994). En viss standardisering krävs dock för att intervjuaren skall kunna jämföra de olika resultaten, det vill säga intervjuaren måste ställa samma frågor vid alla intervjuer. Frågorna har till syfte att undersöka om personer med erfarenhet av systemutveckling anser att PD går att integrera med traditionella systemutvecklingsmetoder och på vilket sätt användarmedverkan representeras i de olika metoder som intervjupersonerna använder vid systemutveckling.

Varje intervju inleds med en kort presentation av arbetet och dess syfte, där begrepp som PD, traditionell systemutveckling och speciellt PD-teknikerna och deras användning beskrivs. Denna presentation skall vara lika för alla intervjupersoner så att de har samma referensram då intervjun börjar. Vidare skall även frågorna (bilaga 1) vara exakt lika för intervjupersonerna med utrymme för diskussion och följdfrågor. De första frågorna behandlar vilken roll intervjupersonen har haft vid systemutveckling, vilken metod som används och hur denna är uppbyggd. Samtidigt som de har som funktion att få igång en dialog. Följande frågor ställs för att besvara detta:

1. Hur länge har ni arbetat med systemutveckling?
2. Vilken roll eller vilka roller har ni haft i systemutvecklingsarbetet?
3. Vilka metoder har ni använt vid systemutveckling och hur är dessa klassificerade (traditionell, objektorienterad etc.)?

Genom dessa frågor får man en bild över hur erfaren intervjupersonen är och vilken roll personen haft i olika systemutvecklingsprojekt för att säkerställa att det är rätt person som intervjuas. Den tredje frågan kommer att besvara vilken typ av metod som används och utifrån detta beslutas vilken metod som skall ligga till grund för resterande frågor.

Efter fråga tre kommer frågor om användarmedverkan i denna metod, vilken roll användarna har, till vilket syfte, när och hur användarna är med i systemutvecklingsprocessen. Detta för att kunna jämföra användarmedverkan i traditionella metoder med användarmedverkan i SSADM för att undersöka om det går att integrera dessa traditionella metoder på samma sätt som med SSADM. Dessa frågor är:

4. Hur är metoden uppbyggd, faser, tekniker etc.?
5. Ger metoden stöd för användarmedverkan?
6. Vilken är användarnas roll vid systemutvecklingen med er metod?
7. När i systemutvecklingsprocessen deltar användarna och på vilket sätt?
8. Vad är syftet med användarmedverkan för er i systemutveckling?
9. Vilka tekniker används för att få information ifrån användarna?
10. Anser ni att det sätt att representera användarna i er metod är bra eller skulle det kunna göras på något annat sätt, i så fall hur?

Fråga fyra har för avsikt att undersöka hur den metod som används är uppbyggd för att kunna jämföra denna med SSADM och för att kunna dra paralleller mellan de olika metoderna. Genom att hitta likheter och skillnader kan man avgöra om PD går att använda på liknande sätt i alla traditionella systemutvecklingsmetoder.

## 4 Metod

Fråga fem till tio behandlar användarna och deras roller i systemutvecklingsprocessen. Avsikten är att få information om användarnas syfte i systemutvecklingen, när och på vilket sätt användarna deltar i processen och vilka tekniker som används för att få information från användarna. Även svaren på dessa frågor kommer att jämföras med SSADM och användas för att undersöka om det är möjligt att integrera traditionella metoder med PD.

Svaren på dessa frågor kan sedan vägas mot litteraturen om användarmedverkan i SSADM och på så sätt användas för att avgöra om PD-teknikerna även går att i andra traditionella metoder än SSADM.

Intervjun avslutas med att fråga om PD kan integreras med traditionella metoder:

11. Tror ni att PD kan användas inom traditionell systemutveckling? Varför/varför inte?

Svaret på frågan kommer att ge en indikation om det är möjligt att genomföra denna integration i verkligheten. Vidare ger frågan även svar på varför det inte skulle vara möjligt respektive varför det skulle vara möjligt att använda PD inom traditionell systemutveckling.

Resultatet av dessa delmål tillsammans med intervjuerna kommer sedan att användas för att besvara huvudmålet i arbetet, det vill säga om det går att integrera PD-tekniker med traditionella systemutvecklingsmetoder.

## 5 Participatory design

Framtidsseminarier och kooperativ prototyping representerar tekniker som är mer användardrivna än vad traditionella tekniker är. Teknikerna är utformade för att kunna låta användarna ta aktiv del i systemutvecklingsprocessen och lättare kunna uttrycka sina krav på det nya informationssystemet. Nedan följer en beskrivning utav de två valda teknikerna framtidsseminarium och kooperativ prototyping.

### 5.1 *Framtidsseminarium*

Denna teknik utvecklades i Tyskland 1987 av Jungk och Mullert. Ursprungligen användes den vid exempelvis stadsplanering och miljöfrågor. Tekniken används för att generera idéer för framtida aktiviteter och initiera aktionerna i arbetet och hur dessa skall implementeras. Tekniken är också designad för att kunna inkludera människor som i vanliga fall inte arbetar med beslutsfattande, det vill säga de människor i organisationen som inte har till uppgift att fatta avgörande beslut om organisationen, och har visat sig vara mycket effektiv för att involvera användarna i systemdesign (Greenbaum och Madsen, 1993).

Framtidsseminarier ger användarna av informationssystemet möjlighet att på ett enkelt sätt uttrycka sina åsikter och funderingar om det nya informationssystemet (Greenbaum och Madsen, 1993). Ett framtidsseminarium består generellt av två seminarieledare och upp till 20 seminariedeltagare. Ledarna skall försöka se till att alla på seminariet får tala lika mycket. De skall även se till att alla följer med i diskussionen genom att låta deltagarna skriva ner sina idéer som korta uttalanden på papper som sätts upp på en vägg (Kensing och Madsen, 1991).

Ett framtidsseminarium är indelat i tre faser; kritisk fas, fantasifas och implementationsfas. Den kritiska fasen har till syfte att lyfta fram det nuvarande arbetet, fantasifasen låter deltagarna föreställa sig om och hur arbetsplatsen skulle kunna se annorlunda ut och slutligen fokuserar implementationsfasen på vilka resurser som behövs för att göra förändringarna (Kensing och Madsen, 1991).

#### 5.1.1 Den kritiska fasen

Den kritiska fasen påminner i stort om en strukturerad brainstorming där fokus ligger på de nuvarande problemen i arbetet. Deltagarna får rikta kritik mot arbetet och lyfta fram problem samtidigt skall dessa antecknas och sättas upp på en vägg så att alla uttalanden samlas på ett ställe. Deltagarna skall ha samma tid att prata på för att alla skall ha samma möjligheter. Viktigt är att deltagarna inte skall behöva försvara eller argumentera för eller emot sina idéer. Efter att kritiken kommit fram delas denna upp i ett antal teman, vilka sedan diskuteras i mindre grupper (Greenbaum och Madsen, 1993). Exempel på kritik kan vara problem med orderhantering, dåligt stöd och ingen kontroll över inventering etc. (Kensing och Madsen, 1991). Resultatet av dessa diskussioner redovisas sedan av deltagarna. Här är det viktigt att seminarieledarna uppmuntrar deltagarna att fokusera på problemen och inte på hur dessa problem skall lösas (Greenbaum och Madsen, 1993).



### 5.1.2 Fantasifasen

I denna fas skall deltagarna komma fram till hur arbetet skall se ut i framtiden. Även denna fas liknar brainstorming där deltagarna använder sin fantasi för att uttrycka hur det skulle kunna se ut i framtiden detta gäller både själva arbetet och de framtida datorapplikationerna. Viktigt i denna fas är att ingen idé skall anses vara för extrem, exempel på idéer kan vara åtkomst till en databas hemifrån, se över databasen, förenkla ordersystemet eller inhandla ny kaffekokare. För att kunna gradera vilka krav som är viktigast måste deltagarna på något sätt begränsa antalet idéer, detta kan göras genom att varje deltagare får ranka kraven och på så sätt få en bild av vad som är viktigast. Resultatet diskuteras sedan för att komma fram till alternativa lösningar (Kensing och Madsen, 1991).

### 5.1.3 Implementationsfasen

Implementationsfasen börjar med att man går igenom vad som uppkommit från föregående fas. Efter detta diskuteras hur man skulle kunna realisera de krav som framkommit, exempelvis kan problemet med bokningssystemet lösas genom att införa ett nytt informationssystem och vilka funktioner skulle detta system i så fall skulle behöva ha (Kensing och Madsen, 1991). Implementationsfasen fungerar som en bro mellan visionen och verkligheten (Greenbaum och Madsen, 1993). Under denna fas måste deltagare och seminarieledare vara medvetna om några viktiga element som måste tas under övervägande. Enligt Greenbaum och Madsen (1993) är dessa element:

- Hinder och begränsningar.
- Möjligheter.
- Ta ner visionerna från fantasifasen på jorden.
- Prioritera kraven.
- Sätta upp en handlingsplan.

Denna fas avslutas med att deltagarna tillsammans tar fram en handlingsplan över vad som behövs för att kraven skall realiseras (Kensing och Madsen, 1991). För varje del planen inkluderar skall dokumentation göras om vad som skall göras, när och hur det skall göras. Slutligen samlas denna information till en rapport (Greenbaum och Madsen, 1993).

### 5.1.4 Fördelar och nackdelar

Framtidsseminarier hjälper systemutvecklarna att komma fram till målet med det nya systemet genom att fokusera på det arbete som människorna i organisationen utför i stället för att fokusera på tekniska och ekonomiska aspekter. Detta tillvägagångssätt gör det möjligt för användarna att aktivt medverka i projektet och att delge sin kunskap till systemutvecklarna och på detta sätt få fram de "riktiga" problemen i arbetet. Nackdelar med framtidsseminarium är att det är tidskrävande. Vidare är det svårt att välja ut deltagare och att arbeta med dessa utan att seminarieledarna påverkar eller manipulerar dem under arbetet (Kensing och Madsen, 1991).

### 5.2 *Kooperativ prototyping*

Traditionella ansatser av prototyping, exempelvis när prototypen blir det färdiga systemet, innefattar designernas perspektiv och tar föga hänsyn till användarna i systemutvecklingsprocessen. En annorlunda ansats inom prototyping är kooperativ prototyping som har sina rötter i explorativa ansatserna, vilket innebär att man bygger simulerade prototyper av systemet. Skillnaden är att i kooperativ prototyping sker utvecklingen i samarbete mellan användare och designers istället för att designers sköter utvecklingen och tar hjälp av användaren för att ta fram krav (Bødker och Grønbæk , 1991).

Kooperativ prototyping är en ansats där syftet är att skapa en designprocess där både användare och designers deltar aktivt och kreativt med sina olika kvalifikationer. För att göra denna process genomförbar måste designers på något sätt låta användarna uppleva arbetsliknande situationer med framtida datorapplikationer. Detta innebär att användarna måste komma i kontakt med den nya teknikens möjligheter, vilket kan göras genom att simulera arbetssituationer eller att använda verkliga situationer. Kraschar systemet i någon av dessa situationer kan användare och designers diskutera vad som hänt och varför, direkt då felet uppstått. Det går då snabbt och enkelt att förbättra designen eftersom designerna direkt kan angripa problemet då användarna är med och direkt kan utvärdera prototypen. För att prototypen skall bli så bra som möjligt bör användarna få känna att de har kontroll över den under en relativt lång period. Under denna tid skall designers vara till stöd för att förbättra prototypen under det att tiden går. En idealisk kooperativ prototyping skall utföras i små grupper av användare och designers och det skall finnas åtkomst av datorbaserade verktyg för snabb utveckling och modifiering av prototypen (Bødker och Grønbæk , 1991).

För att kooperativ prototyping skall fungera är det viktigt att skapa en förståelse för målet med processen, vad som skall utvecklas och prototypens roll genom designprocessen. Vidare måste man lösa olika organisatoriska problem innan arbetet med prototypen kan påbörjas (Bødker och Grønbæk , 1991).

#### 5.2.1 **Fördelar och nackdelar**

Prototyping tillämpas med fördel i analysfasen av ett systemutvecklingsprojekt där den kan komplettera de abstrakta beskrivningar av verksamheten i form av t.ex. informations- och affärsprocessbeskrivningar som ofta tas fram där. En prototyp kan bidra till ökad förståelse för de praktiska konsekvenserna av modellerna för användarnas arbete. Detta gäller speciellt om analysen av verksamheten har resulterat i att den skall förändras, t.ex. att arbetet skall genomföras på ett nytt och mer effektivt sätt. Det kan då vara svårt att föreställa sig konsekvenserna för det dagliga arbetet. På så vis kan prototypen dels bidra till att stämma av specifikationerna och förankra dessa hos användarna, dels ge rekommendationer till det framtida utvecklingsprojektet. Tekniken kan dock även användas för prototyputformning av det slutgiltiga gränssnittet (Rosengren och Wingstedt, 1993).

Ytterligare fördelar är att de blivande användarna genast kan ge sina synpunkter och systemutvecklarna kan då snabbt korrigera felen (Flensburg och Friis, 1999). Vidare är kooperativ prototyping en teknik som kan förbättra kvaliteten av ett system utifrån användarnas synvinkel, eftersom användarna lättare kan uttrycka sina önskemål genom att använda prototyping (Bødker och Grønbæk , 1991). Nackdelen är att prototypen

## 5 Participatory design

endast kan ses som en kravspecifikation och det kan ta lång tid innan det färdiga systemet kan tas i drift. Risken är då stor att organisationen hinner förändras innan det nya systemet tas i drift och blir då inaktuellt innan det är färdigt. En annan risk är att användarna blir otåliga i väntan på det nya systemet, vilket kan leda till konflikter. Dock finns det sätt att lösa dessa problem. Ett sätt är att tillfälligt ta prototypen i drift (Flensburg och Friis, 1999). Detta är även en fördel då problem med prototypen uppstår ser man felet direkt och kan rätta till det (Cherry och Macredie, 1999).

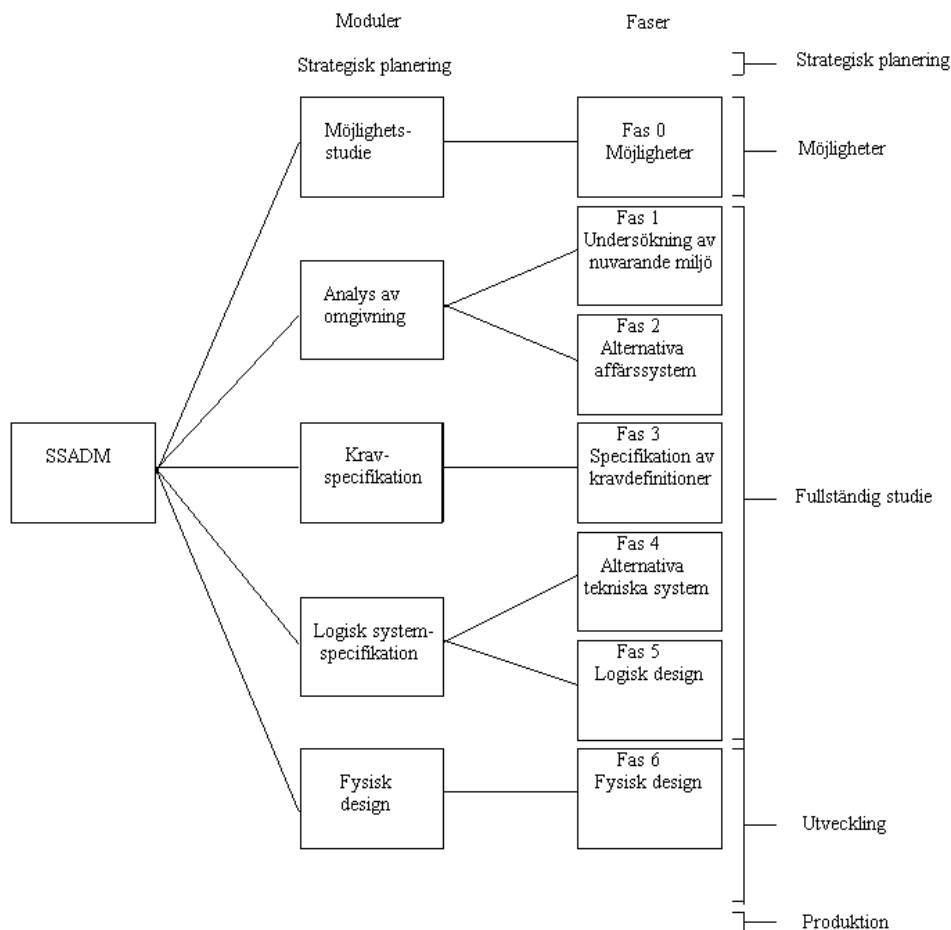
Vid prototyping fokuserar man mest på innehållet och funktionen hos systemet. Eftersom det är en iterativ process kan det vara svårt att fastställa dels hur lång tid det tar att utveckla det nya systemet och dels hur mycket det kommer att kosta. Detta är en nackdel då det kan leda till att företag inte vågar satsa för mycket på kooperativ prototyping (Flensburg och Friis, 1999).

## 6 Systemutvecklingsmetoden SSADM

Metoden SSADM är ett sätt att organisera analys och design för ett nytt informationssystem, där målet är att leverera ett datoriserat informationssystem.

SSADM består av fem moduler, sex faser och 33 steg och olika tekniker, vilka tillsammans beskriver arbetet för design och analys under systemutvecklingen (Figur 1). Modulerna är följande:

1. Undersökning av möjligheter
2. Analys av omgivning
3. Kravspecifikation
4. Logisk systemspecifikation
5. Fysisk design



Figur 1- Metoden SSADM (Downs, 1992 sid 3)

Även om SSADM är en strukturerad metod krävs det att den anpassas efter olika situationer inom projektet.

Huvuduppgifter för metoden är att:

1. Fastställa vilket stöd användarna skall få av det framtida systemet.

### 2. Designa det önskade systemet.

SSADM lägger inte tyngdpunkten på systemförvaltning eller själva programmeringen av systemet dock måste metoden interagera med dessa delar.

SSADM består av aktiviteter och produkter. Aktiviteterna kan beskrivas på två sätt; dels *när* någonting skall göras och dels *hur* någonting skall göras. Produkterna i SSADM innefattar *vad* som skall levereras. Metoden består av en hierarki av moduler som är indelade i faser, steg och uppgifter. Inom denna struktur används olika tekniker för att producera produkter eller uppdatera produkter. Alltså fungerar all indata och utdata som produkter, vilka sedan dokumenteras väl och blir till antalet runt 100 stycken.

Nedan följer en detaljerad beskrivning av systemutvecklingsmetoden SSADM, dess tillvägagångssätt, tekniker och användbarhet. Teknikerna behandlas i ett separat kapitel för att kunna förstå beskrivningen av de olika faserna i kapitel 6.2.

### 6.1 Tekniker i SSADM

Teknikerna i SSADM används oftast parallellt med varandra. De tekniker som främst vänder sig till användarna är dataflödesmodellering och kravdefinition. Inom SSADM finns det en mängd olika tillvägagångssätt för att finna fakta och information som behövs i de olika teknikerna och för att bygga det nya informationssystemet. Dessa är följande:

- Intervjuer
- Analysera dokument från det nuvarande informationssystemet
- Enkäter
- Arbetsdagböcker specialdesignade för systemundersökning
- Skugga användare eller utföra användarnas arbetsuppgifter
- Observation
- Prototyping
- Brainstorming med användarna
- Använda analytikernas egen erfarenhet från liknande systemutvecklingsprojekt

Av ovanstående tillvägagångssätt är prototyping och brainstorming begrepp som ofta används inom PD. Skillnaden är dock att då de används inom PD är det användarna som styr brainstormingen och utvecklar prototyperna medan systemutvecklarna fungerar som stöd, exempelvis genom att se till att alla får prata lika länge i brainstormingen. I traditionell systemutveckling leds deltagarna, exempelvis användare eller chefer, i brainstormingen av en systemutvecklare och prototyperna utvecklas av systemutvecklare som sedan använder användarna för att testa prototyperna.

Vilket alternativ som används beror på hur organisationen är uppbyggd och vilka människor som är inblandade (Downs, 1992). Det är i litteraturen mycket oklart när dessa olika tillvägagångssätt för att få fram information från användarna används och den beskriver inte heller hur det görs eller med vilket syfte de används. Exempelvis är det mycket svårt att avgöra skillnaden mellan observation och att skugga användare när detta inte finns beskrivet i metoden.

Nedan följer en kort beskrivning av de tekniker där användarna på något sätt är inblandade, när teknikerna används och till vilket syfte.

### 6.1.1 Kravdefinition

Tekniken innebär att identifiera och beskriva vad som krävs av det nya informations-systemet. Syftet är att få kunskap om organisationen och vad den behöver i framtiden. Analytikern skapar en kravkatalog där inte bara kraven beskrivs utan även frågar vem som ställer kravet, varför och hur viktigt det är. Tekniken används i alla modulerna. I modulen för möjligheter används tekniken för att undersöka det nuvarande systemet, hitta problem med detta och identifiera krav på det framtida systemet. I modulen för kravanalys används tekniken för att etablera ett analysramverk, undersöka och definiera krav, härleda en logisk syn på nuvarande tjänster och att definiera alternativa affärssystem. Kravspecifikation är nästa modul där tekniken används. Detta för att definiera systemprocesser och utveckla datamodell, härleda systemfunktioner, utveckla specifikationsprototyper och bekräfta systemets delmål. Vidare används även tekniken i specifikation av logiskt system och i den fysiska designen. Detta för att definiera alternativa tekniska system, definiera användardialoger och att kontrollera att den fysiska designen stämmer överens med de krav som finns på systemet (Meldrum, 1993).

### 6.1.2 Dialogdesign

Denna teknik innefattar två deltekniker, dels identifiering av dialoger som börjar i fas 1 och slutar i fas 3, dels dialogdesign som sker i fas 5. Båda delteknikerna behandlar dialoger mellan användarna och informationssystemet men den första koncentreras på att utarbeta vilken användare som använder vilken funktion. Detta görs genom att lista användarna och deras roller mot en funktionsmatris. I den andra deltekniken, dialogdesign, skapas I/O-strukturer och dessa element grupperas och sökvägar till de olika grupperingarna identifieras. Syftet med tekniken är att slutligen utveckla kommando- och menystrukturer, dock behandlar inte tekniken gränssnittet för menyerna utan endast processerna bakom. Tekniken ger riktlinjer för att identifiera användarnas krav och dess sammansättning till funktionsdefinitioner. Den beskriver alltså hur data-processer kan mappas med användarnas arbetsmönster (Downs, 1992). För att kontrollera att strukturerna är korrekta bör konsultation med användarna ske.

### 6.1.3 Dataflödesmodellering

Ett dataflödesdiagram visar flödet av data inom organisationen, det vill säga vart data tar vägen i ett informationssystem. Tekniken är grafisk vilket gör den enkel att förstå för användarna. Tekniken är ett bra och enkelt sätt att modellera ett informations system, vilken gör den användbar under analysfaserna. Modellen ger ett bra stöd för att identifiera systemets gränser, existerande aktiviteter och är mycket ett användbart verktyg vid intervjuer av användarna. Modellen används även för att identifiera användarnas krav, vilka fungerar som bas för det nya systemet. Tekniken är ett bra verktyg för att kommunicera med användarna då den är enkel att förstå och använda (Downs, 1992). Dataflödesmodellering används tidigt i utvecklingsprocessen, i modulerna studie av möjligheter, krav analys och kravspecifikation (Meldrum, 1993).

### 6.1.4 Logisk datamodellering

Denna teknik bygger på att skapa en logisk datastruktur, vilket är likartat med ett ER-diagram i andra ansatser. Tekniken innefattar att identifiera entiteter och relationer mellan dessa. För att få fram dessa fakta involveras både utvecklare och användare i processen och sker alltid parallellt med dataflödes modellering vilken är enklare för användarna att förstå. Diagrammen förfinas sedan under utvecklingens gång genom diskussion med användarna. Tekniken används i alla moduler i SSADM (Meldrum, 1993).

### 6.1.5 Alternativa affärssystem

Detta är en produkt, en fas och en teknik, vilken har till syfte att bestämma den huvudsakliga omgivningen av informationssystemet som utvecklas. Tekniken hjälper systemutvecklarna att ta fram några alternativa lösningar på affärssystem. Alternativen diskuteras sedan med användare och finansörer varav ett alternativ slutligen väljs. Viktigt i denna teknik är att utvecklarna inte leder diskussionen utan att användare och finansörer får känna att de äger systemet (Downs, 1992). Tekniken används i de tre första modulerna (Meldrum, 1993).

### 6.1.6 Funktionsdefinition

Funktioner definieras för att representera de tjänster och det stöd användarna vill skall grupperas tillsammans. Det görs för att den underliggande strukturen på informationssystemet skall vara meningsfull för det arbete som skall utföras. Strukturen bygger på de krav som användarna satt upp för det nya systemet. En funktion är alla processer som krävs för att indata skall flöda genom systemet sett från användarnas synvinkel, härav är det mycket viktigt att diskutera funktionerna med användarna för att få fram rätt information då dessa sitter på detaljerad information om aktiviteter (Downs, 1992). Genom att konsultera användarna hjälper de analytikern att validera och identifiera funktionerna. Denna teknik används i modulen för kravspecifikation, logisk systemspecifikation och fysisk design (Meldrum, 1993).

### 6.1.7 Relationsdataanalys

Denna teknik handlar om att skapa relationer mellan data. Analysen görs för att kontrollera att relationerna är korrekta, att entiteterna innefattar enkla strukturer och att användarnas syn på data är konsistent med utvecklarnas (Downs, 1992). Genom att använda relationsdataanalys fås ökad förståelse av den data i systemet som skall undersökas. Tekniken sker parallellt med logisk datamodellering och används i möjlighetsstudien, kravanalysen och kravspecifikationen.

### 6.1.8 Alternativa tekniska system

Liksom den ovan nämnda tekniken alternativa affärssystem handlar denna teknik om att ta fram några alternativ på tekniska system varav ett skall väljas. Även här är det viktigt att låta användare och sponsorer göra valet med hjälp från systemutvecklarna inte tvärt om (Downs, 1992). Tekniken används i modulen logiska systemspecifikation och i den fysiska designen (Meldrum, 1993).

### **6.2 Faser i SSADM**

I följande kapitel beskrivs SSADMs olika faser och steg. I kapitel används Downs (1992) som referenslitteratur (se Bilaga 2).

#### **6.2.1 Undersökning av möjligheter**

Projektstyrelsen är ansvarig för studien av möjligheter. De måste ta reda på om organisationens resurser skall satsas i projektet eller inte och, om så är fallet, vilka resurser som skall användas. Arbetet för att göra detta ligger utanför SSADM härav varierar detta arbetet i olika projekt men generellt behandlas följande:

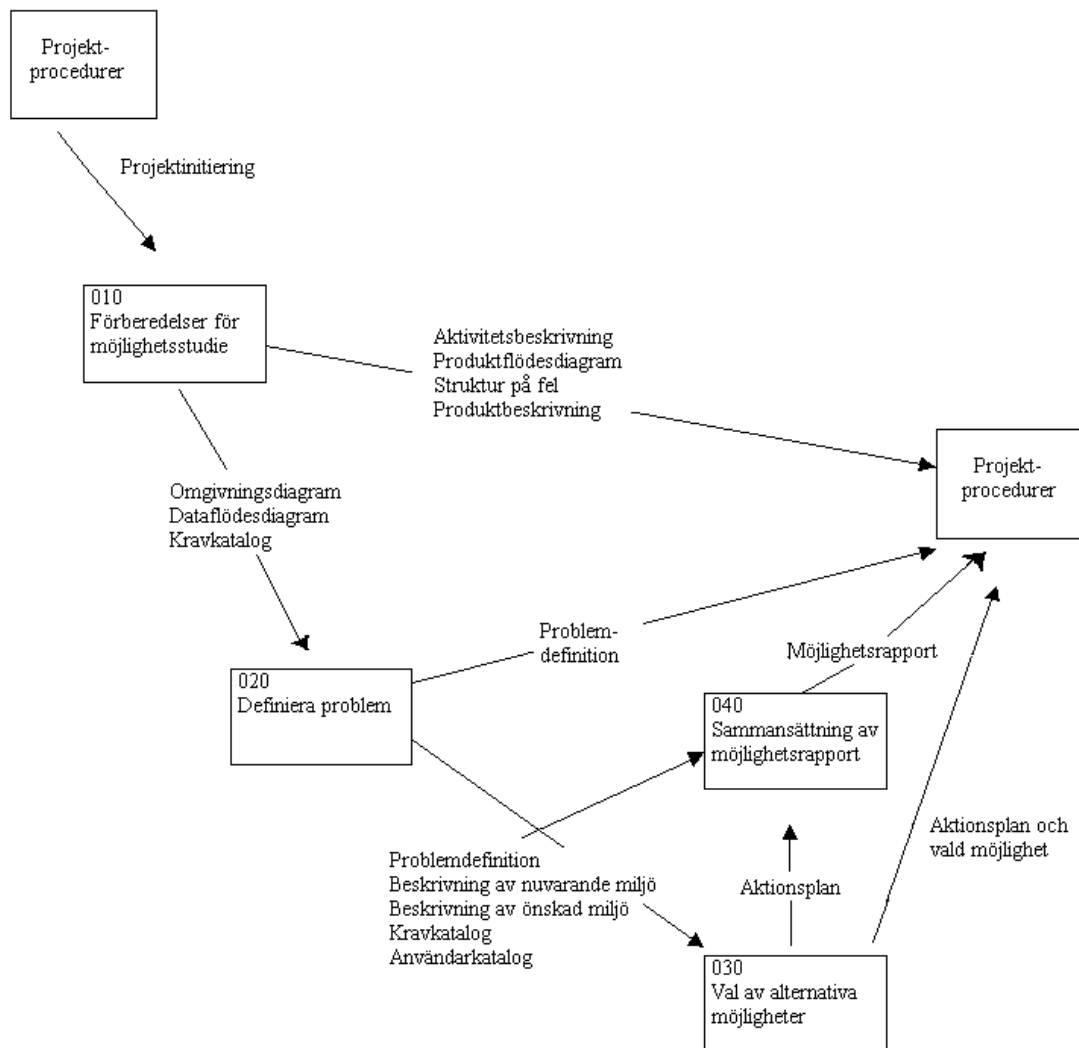
1. Projektinitiering och styrning, vilket inkluderar att få stöd för själva studien och tillsättande av resurserna (inkluderat människor).
2. Användar- och ledningsinvolvering för att försäkra sig om att de som är inblandade i utvecklingen är involverade och att passande procedurer följs.
3. Komma överens om problemsituationen och utnämna prioriteringar.
4. Utvärdera olika alternativ, speciellt finansiella aspekter och relationerna mellan människornas arbete och organisationen.

Trots att arbetet ligger utanför metoden, vilket innebär att det skulle vara möjligt att utföra denna modul på ett annat sätt med en annan metods motsvarighet till SSADMs tillvägagångssätt på undersökning av möjligheter, är det en mycket viktig del i utvecklingsarbetet. SSADMs roll i denna modul är att studera området som skall undersökas, för att kunna ge information om behov inom organisationen och hur dessa behov skulle kunna lösas genom ett datoriserat informationssystem. Detaljnivån behöver inte vara djupare än att man skall kunna fatta beslut.



### 6.2.1.1 Fas 0 - Möjligheter

Modulen består av en fas och fyra steg (se Figur 2).



Figur 2 – Möjlighetsstudie (Downs, 1992 sid 21)

#### Steg 010 - förberedelse för möjlighetsstudien

Detta steg skapar en bas på vilken de andra stegen skall byggas. Det som görs är att dokumentera en beskrivning av det nuvarande systemet och de kända krav som finns. Slutligen skall en överenskommelse med projektstyrelsen ske för att gå vidare med projektet. De tekniker som används är dataflödesmodellering, logisk datamodellering och kravdefinition.

#### Steg 020 - Definition av problem

Målet med detta steg är att förstå det huvudsakliga problemet och de möjligheter som finns med ett datoriserat informationssystem för att möta detta problem. Det är inte meningen att man här skall gå på djupet med det nuvarande systemet och dess

svårigheter eller människornas förhoppningar om framtiden. Användarrollen är dock nödvändig för att kontrollera att systemutvecklarna förstått det nuvarande systemet och kraven på det nya. Uppgifterna i detta steg skall ge ett öppet tankesätt om det framtida informationssystemet genom att studera det nuvarande mer detaljerat. Produkterna i detta steg inkluderar en beskrivning av nuvarande och framtida omgivning. De uppgifter som utförs i detta steg är att ge en översikt av det önskade systemet med hjälp av dataflödesdiagram och logisk datastruktur. För att få en klar blick över det önskade systemet utökas dessa beskrivningar efter hand och problem dokumenteras i kravkatalogen. Vidare skall potentiella användare av det krävda systemet dokumenteras i en användarkatalog. Nya krav från dessa användare skall tas fram genom tekniken diskussionsdialog och dokumentera dessa i kravkatalogen dessa krav skall sedan slås samman med problemdefinitionen. Slutligen skall denna definition bekräftas av projektstyrelsen.

### **Steg 030 - val av möjliga alternativ**

Målet med detta steg är att ta fram möjliga alternativ på fortsatt projektarbete. De riktlinjer man utgår ifrån är:

- Kostnader för att anta projektet och vilka uppoffringar som kan bli aktuella.
- Förändringar i företaget och för människor som är involverade.
- Risker förknippade med säkerhet, granskning och konfidentialitet.
- Livsdugligheten i projektet.

De uppgifter som skall utföras i detta steg är fastställa vilka krav som måste mötas. Ta fram en översikt på alternativa affärssystem och tekniska system där fokus ligger på olika skillnader mellan alternativen. Vidare skall projektplaner för varje alternativ utformas och avgöra vilka alternativ som föredras för att sedan presentera alternativen för projektstyrelsen och välja ett. Slutligen skall en aktionsplan utvecklas för det valda alternativet. Tekniker som används här är dataflödesmodellering, logisk datamodellering, affärssystemsväl och tekniska systemväl.

### **Steg 040 - sammansättning av möjlighetsrapport**

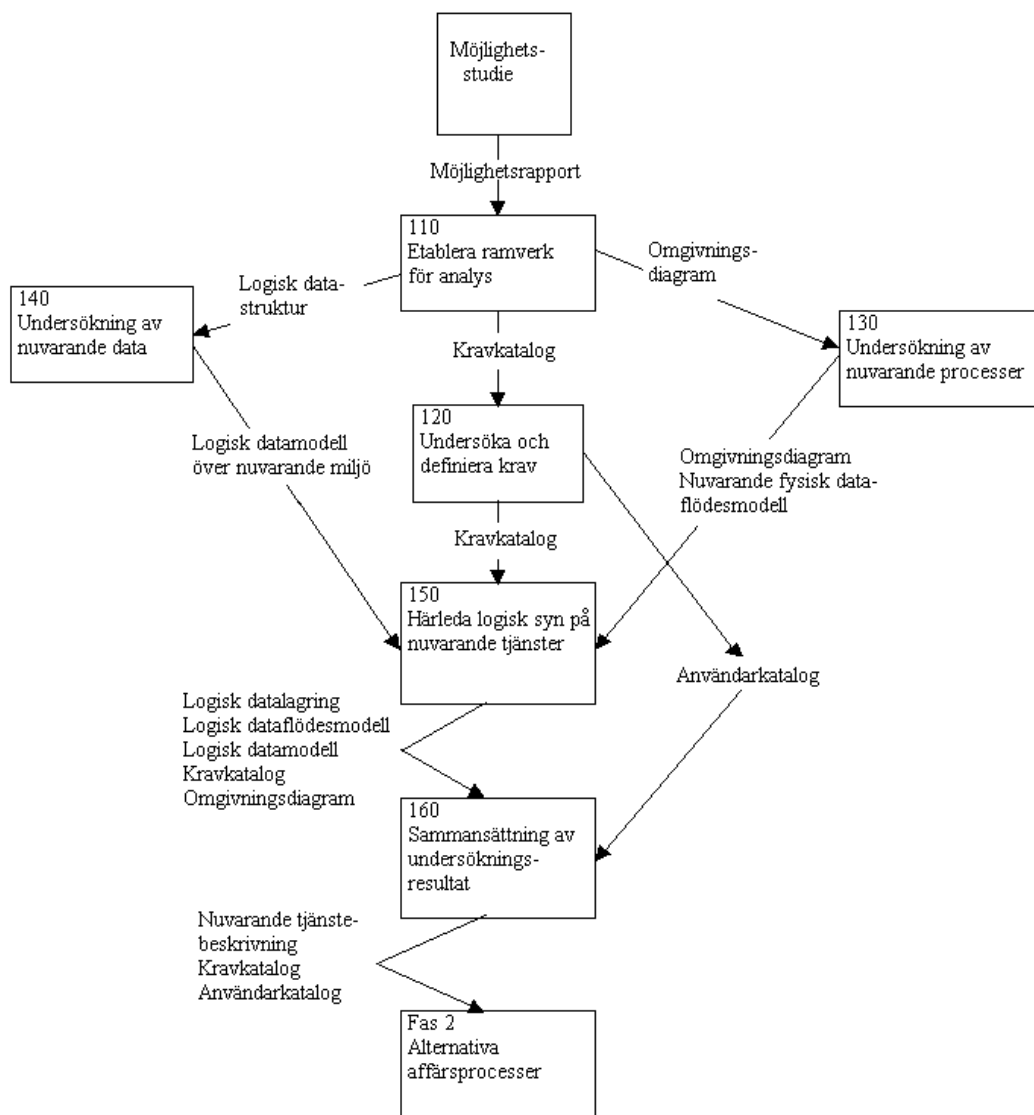
Varje modul i SSADM avslutas med ett sammansättningssteg. Syftet är att kontrollera att det som åstadkommit har gjorts på ett tillfredsställande sätt och att publicera detta resultat. Stegets primära syfte är kvalitetskontroll. Detta steg gör det möjligt att kontrollera varje produkt i modulen och på så sätt upptäcka svårigheter och lösa dessa. Indata är produkterna från de tidigare stegen och resultatet blir utdata i form av en rapport över möjlighetsstudien.

### **6.2.2 Kravanalys**

Denna modul består av två faser, den första, undersökning av nuvarande miljö, är en undersökning av nuvarande omgivning. Denna fas innefattar att få kunskap om affärsområdet och utveckla en logisk bild av nuvarande aktiviteter och framtida behov. Den andra fasen, affärssystemalternativ, utvärderar ett antal alternativa handlingslinjer varav en skall väljas av ledningen.

### 6.2.2.1 Fas 1 - undersökning av nuvarande omgivning

Denna fas ger grunden till fortsatt systemutvecklingsarbete genom att fokusera på det existerande informationssystemet och människorna som skall använda det liksom deras framtida behov (se Figur 3).



Figur 3 - Undersökning av nuvarande omgivning (Downs, 1992 sid 29)

Fasen består av sex steg. Det första steget innefattar framtagande av ramverk för analysen av företaget. Beskrivning av det nuvarande informationssystemet skapas i de övriga stegen i denna fas. I steg 150 transformeras beskrivningarna till dess logiska likheter, vilket kan leda till identifiering av förändringar i kraven. Om det inte finns något nuvarande informationssystem kan steg 130 och 140 utelämnas.

Under undersökningens gång kommer kravkatalogen att ständigt uppdateras och kommer att fortsätta uppdateras under hela projektets gång. Det innebär att allt som användarna önskar med det nya systemet kommer att dokumenteras. Detta inkluderar funktionalitetskrav, service som informationssystemet ger och icke-funktionella krav. Kraven skall också organiseras och diskuteras med relevanta användare, vilket gör det

lättare att binda kraven till affärsaktiviteter. Fasen avslutas med att samla produkterna i undersökningen, vilka ligger till grund för att beskriva alternativa affärssystem.

### **Steg 110 - etablera ramverk för analysen**

Det är i detta steg som metoden verkligen börjar användas och det förutsätter att ett förarbete finns som grund. Syftet med detta steg är att se över om det pågående arbetet är till tillfredsställelse och att avgöra hur projektet skall fortsätta. Det är i detta steg som användarna får sin första kontakt med projektet, det är även i här som projektgruppen avgör om det finns tillräckligt med resurser för att fortsätta projektet eller om det skall avslutas.

Indata i detta steg är det som tagits fram i den första modulen. De uppgifter som sedan följer är att se över denna indata och uppdatera denna om det behövs. Vidare skall framtida användare och analysområden identifieras och hur denna analys skall utföras. Slutligen skall man även komma överens om vilken metod som skall användas för vidare arbete. Tekniker som används i detta steg är dataflödesmodellering, dialogdesign, logisk datamodellering och kravdefiniering.

### **Steg 120 - undersöka och definiera krav**

Kravkatalogen skapas i detta steg, om en sådan inte redan existerar från den första modulen. Katalogen är en lista av krav som kommer att bli den drivande kraften i projektet. Slutligen skall detta steg bidra med en lista på användare och deras aktiviteter, så att alla potentiella användare kan bli konsulterade vid senare steg. Det som skall utföras är framtagning av information om det nuvarande systemet och dess problem. Vidare skall potentiella användare dokumenteras i användarkatalogen. Slutligen skall man väcka hopp hos användarna inför det nya systemet och göra en prioritering av kraven. Tekniker att använda är dialogdesign och krav definition.

### **Steg 130 - undersöka nuvarande processer**

Detta steg är helt enkelt dokumentation av det existerande informationssystemet och dess processer, vilket görs med hjälp av dataflödesdiagram, och dess associerade detaljer. Denna information förs sedan in i dataflödesmodellen. Av erfarenhet är det i detta steg som användarna blir entusiastiska då felaktigheter i de nuvarande processerna och en förbättring av dessa är målet med systemutvecklingen.

### **Steg 140 - undersökning av nuvarande data**

Detta steg sker parallellt med steg 120 och 130 och använder den information som erhålls i de stegen. Här identifieras de huvudsakliga entiteterna och detaljer om dess innehåll och nycklar tas fram. Detta är en relativt svår uppgift då alla måste enas om vilka attribut respektive entitet har etc. Den detaljerade information man får fram dokumenteras i datakatalogen, vilken innehåller all information om data på denna nivå. Uppgifter som måste göras är att se över den logiska datastrukturen, det vill säga rita om och utöka om det är nödvändigt, dokumentera information om entiteter, relationer och attribut dokumenteras och slutligen lägga till nya krav som framkommer. Tekniker som används i detta steg är logisk datamodellering och relationsdataanalys.

### **Steg 150 - härleda en logisk syn på nuvarande tjänster**

I detta steg sammanlänkas det arbete som hittills har gjorts om det nuvarande systemet. En av orsakerna till detta är att kontrollera att allt är konsistent. Syftet med steget är att konvertera den fysiska dataflödesmodellen till en logisk dataflödesmodell genom att eliminera externa fysiska faktorer, dubletter och redundans. Därefter dokumenteras

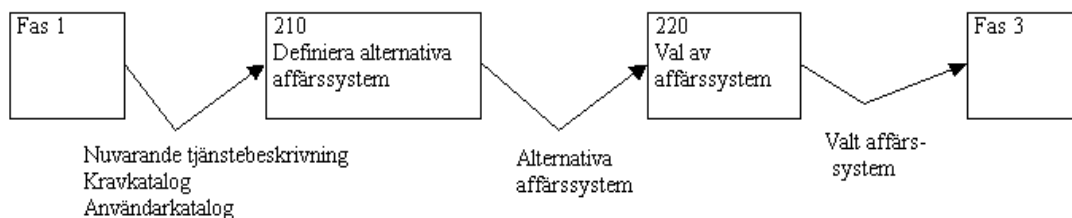
de fysiska kraven i kravkatalogen. Det är här av vikt att dokumentera vilka beslut som fattats och varför om situationen skulle behöva förändras. Det blir då lättare att se vad man har gjort i en viss situation och därav lättare att hantera förändringen. De uppgifter som skall göras är göra dataflödesdiagrammen logiska, matcha den logiska datamodellens entiteter med logiska datalagringar. Efter detta skall de elementära processbeskrivningarna kontrolleras mot den logiska datamodellen och slutligen skall det dokumenteras i kravkatalogen. De tekniker som används är dataflödesmodellering och logisk datamodellering.

### Steg 160 - sammansättning av undersökningsresultat

Detta är det avslutande steget i fasen och behandlar kvalitetssäkring. Här kontrolleras att alla nödvändiga produkter tagits fram och att de håller en viss standard.

#### 6.2.2.2 Fas 2 - Alternativa affärssystem

Det finns ett antal aspekter på alternativen som förbereds och väljs i denna fas. Dessa är det designade informationssystemets omfattning, hur det skall integreras med andra informationssystem och vilken karaktär det skall ha. Andra aspekter är själva utvecklingen, hur informationssystemet skall utvecklas och hur detta arbete skall anpassas till annat arbete. Slutligen finns det aspekter så som kostnaden för och vinster av projektet och informationssystemet (se Figur 4).



Figur 4 - Alternativa affärssystem (Downs, 1992 sid 38)

### Steg 210 - definition av alternativa affärssystem

Efter de tidigare stegen finns det nu ett antal olika möjligheter för det önskade logiska systemet, där varje alternativ skiljer sig exempelvis i tillgängliga tjänster, hur det påverkar organisationen, finansiella kostnader och vinster. Dessa alternativ har både fördelar och nackdelar och det gäller att finna det alternativ som är mest lämpat att lösa organisationens problem. De uppgifter som utförs för att få fram det bästa alternativet är att identifiera de krav som måste mötas av någon lösning. Vidare skall ett antal alternativ tas fram och diskuteras ingående med ansvariga användare och efterhand minska antalet alternativ. Slutligen måste kostnaden för respektive alternativ övervägas. Tekniker att ta hjälp av i detta steg är affärssystemalternativ, dataflödesmodellering och logisk datamodellering.

### Steg 220 - val av affärssystem

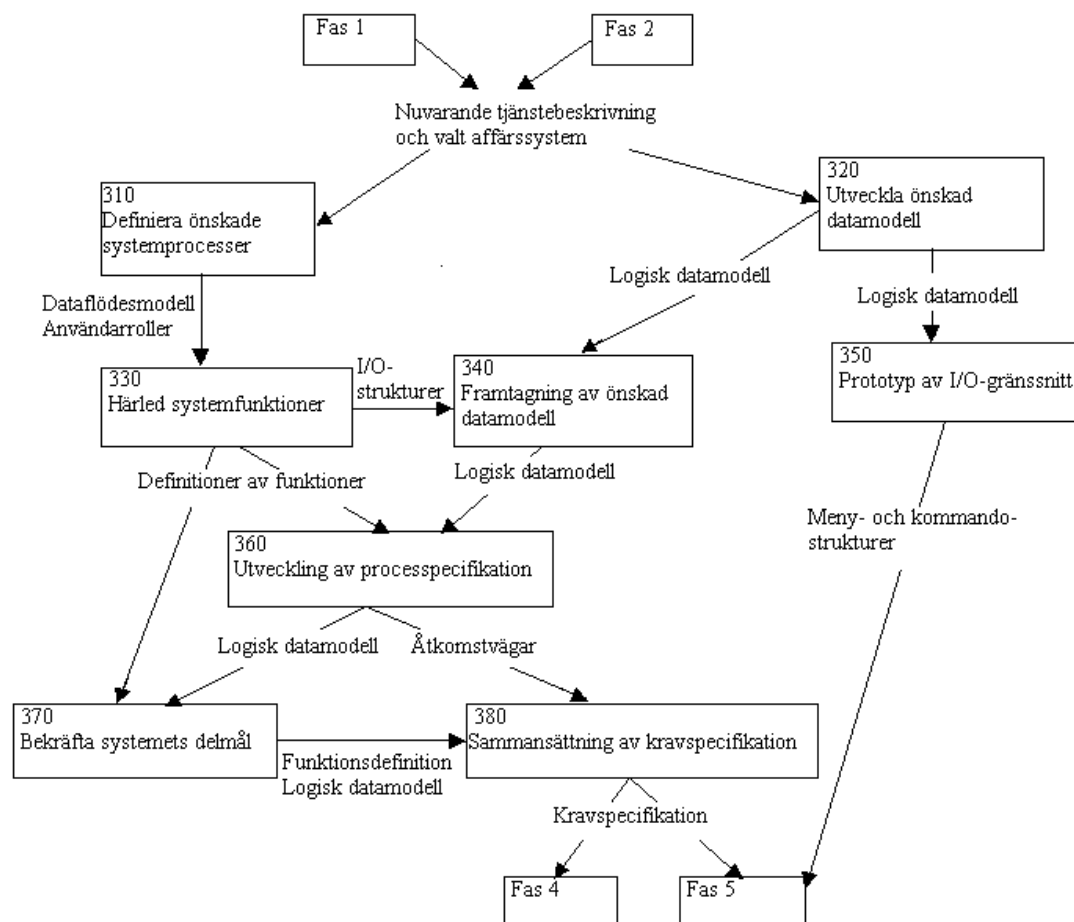
Projektstyrelsen måste fatta beslut om vilken riktning projektet skall ta och hur resurserna skall användas i det fortsatta arbetet. För att göra detta val kan styrelsen dels få råd ifrån systemutvecklarna och dels titta på den dokumentation som gjorts av de olika alternativen. Målet med steget är att styrelsen skall välja ett av de alternativa affärssystemen.

### 6.2.3 Kravspecifikation

Denna modul består av en fas som är mycket komplex, men viktig. I denna fas omvandlas kravanalysen och kravkatalogen till en detaljerad specifikation av de funktioner som det nya systemet skall innehå. Specifikationen är en systembeskrivning vilken innehåller händelser, entiteter och funktioner.

#### 6.2.3.1 Fas 3 - definition av krav

Denna fas tar produkterna från tidigare arbete och använder analystekniker som relationsdataanalys och prototyping för att omarbeta och lägga till produkter för att kunna framställa specifikationen för det nya systemet (se Figur 5).



Figur 5 - Definition av krav (Downs, 1992 sid 44)

#### Steg 310 - definiera behövda systemprocesser

Detta steg utförs parallellt med steg 320 och innefattar en uppdatering av behövda processer, användare och kravkatalogen, vilket innebär en komplett uppdatering av kravkatalogen, dataflödesdiagrammen, detaljerad dokumentation av I/O-strukturen och logiska datamodeller. Vidare skall man kontrollera att den logiska datamodellen för datalagringen motsvarar minst en entitet och att lagringen stöder flödet. Det sista att

utföra i steget är att skapa användarroller för de användare som skall använda det nya systemet och försäkra att rollerna representeras av externa entiteter i dataflödesdiagrammen. De tekniker som används i detta steg är dataflödesmodellering, dialogdesign och kravdefinition.

### **Steg 320 - utveckla krävd datamodell**

Här omarbetas den logiska datamodellen för att möta kraven på det nya systemet. Arbetet innefattar att undersöka varje entitet i kravkatalogen och ändra det nya systemets logiska datamodell så att det stödjer det valda affärssystemet. All data som behövs för det nya systemet kontrolleras så att de finns med i den logiska datamodellen. Slutligen skall alla processer och icke-funktionella krav finnas med i den nya den logiska datamodellen. Tekniker som används för att utföra detta är logisk datamodellering, relationsdataanalys och kravdefinition.

### **Steg 330 - härleda systemfunktioner**

I detta steg bestämmer man funktionerna genom att titta på dataflödesdiagrammen och diskutera processerna i organisationen med användarna. Här börjar man med en formell definition av funktioner och händelser som sedan avslutas i steg 360. I SSADM ses funktionerna från användarnas synvinkel, där funktionerna är en mängd processer, relaterade till varandra, som användarna anser vara av vikt. En produkt från detta steg är en funktionsmatris där det dokumenteras vilka användarroller som hör till olika funktioner, detta görs genom dialog. Målet i detta steg är att identifiera alla funktioner och händelser som skall vara en del av det nya systemet. Funktionerna dokumenteras i en funktionskatalog som inkluderar beskrivning av funktionerna och deras associerade I/O-strukturer. Händelserna dokumenteras i en händelsekatalog som innehåller händelsernas specifikation och diagram som visar vilken effekt de har på entiteterna. Detta är ett mycket viktigt steg för användarna då de kan se deras nya system utvecklas, detta i kontrast till tidigare steg vilka för dem kan ses som endast koncentration på det nuvarande systemet och dess problem.

### **Steg 340 - utöka krävd datamodell**

Här undersöks I/O-strukturerarna och den logiska datamodellen från tidigare steg. För att välja I/O-struktur används relationsdataanalys. Systemutvecklarna plockar ut de strukturer som skall normaliseras, dock kan inte alla strukturer normaliseras då detta är mycket tidskrävande och onödigt. En undersökande analys om det finns avvikelser i strukturerarna måste utföras. Det finns två svårigheter som kan upptäckas. Dels kan det finnas olika beslut fattade om exempelvis någonting är ett attribut till en entitet eller om det är en entitet och dels kan det uppstått missförstånd av innebörden hos vissa ord. Kan inte projektmedlemmarna lösa dessa problem själva måste de ta hjälp av användarna. Tekniker som används i detta steg är relationsdataanalys och logisk datamodellering.

### **Steg 350 - prototyp av I/O-struktur**

Detta steg är valfritt och utförs endast om det anses vara värt att utföras. Steget är koncentrerat till att ta fram ett användargränssnitt som användarna vill använda. Det innebär att man kontrollerar att användarna förstår och menar vad de sagt då de definierat sina synpunkter i steg 330, härledning av systemfunktioner. Det är här projektledningen som beslutar om prototyping eller standardiserade gränssnitt skall användas. De uppgifter som skall utföras är beslut om vilka dialoger mellan användare och systemet som skall användas till prototyping. Vidare skall prototyper göras på

menyer, kommandosekvenser och dialoger. Efter detta skall prototyperna förberedas för test, som sedan utförs på användarna. Slutligen skall resultatet dokumenteras noggrant. De tekniker som används i detta steg är dialogdesign och specifikationsprototyping.

### **Steg 360 - utveckla processspecifikation**

Detta steg är hjärtat i SSADM. Tekniken som används i detta steg är entitets-händelsemodellering och har av två syften. Det första är att tvinga utvecklarna att överväga hur detaljer i det nya systemet kommer att fungera. Det andra är att definiera fundamentala dataprocesser i det nya systemet. Den första uppgiften är att arbeta igenom alla entiteter för att ta reda på vilka händelser som medför att lagrad data måste uppdateras. Där efter är det nödvändigt att kontrollera att det finns definierade funktioner för alla händelser. Då detta är gjort går entiteterna åter igenom två gånger. Den första gången för att skapa entiteternas livscyklar och den andra gången för att lägga till komplexitet som uppstår vid exempelvis interaktion.

Då detaljerna i det nya informationssystemet är etablerade, det vill säga hur händelser påverkar entiteter, dokumenteras detta i ett effektdiagram. Vidare dokumenteras åtkomstvägar, och för detta används logisk datamodellering. Det viktiga med dessa dokument är att de bildar en länk mellan processerna och den logiska databasen.

I detta steg är det nödvändigt att besvara de svårigheter som entitets-händelsemodelleringen ger och gå tillbaka och titta på den logiska datamodellen, funktionsdefinitioner, dataflödesmodellen och tala med användarna.

### **Steg 370 - bekräfta systemets delmål**

I detta steg går systemutvecklarna igenom kravkatalogen för det valda affärssystemet för att kontrollera att alla funktioner kan mötas av det planerade informationssystemet. Om några krav har glömts i tidigare steg är det möjligt att steget måste göras om för att få med kraven i det slutliga systemet. Det skall kontrolleras att alla krav, både funktionella och icke-funktionella, är fullt specificerade och kompletta. Slutligen skall kontroll ske så att systemets logiska datamodell innefattar även behoven av icke-funktionella krav. För att utföra detta arbete används teknikerna kravdefinition, logisk datamodellering och funktionsdefinition.

### **Steg 380 - sammansättning av kravspecifikation**

I detta avslutande steg kontrolleras att alla produkter som skall finnas verkligen finns och att dessa är konsistenta och håller standarden. Slutligen samlas dessa produkter och resulterar i en komplett kravspecifikation.

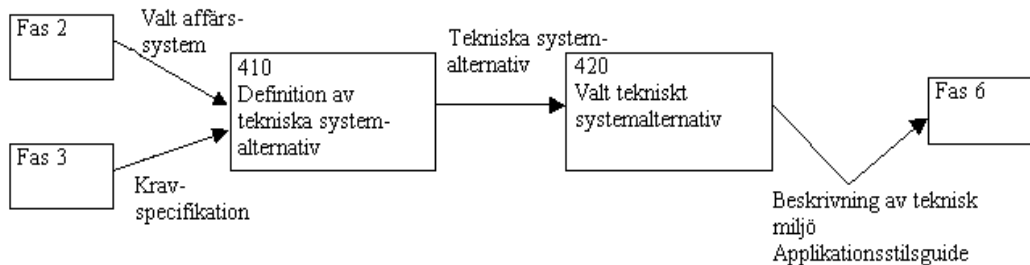
## **6.2.4 Logisk systemspecifikation**

Denna modul innehåller två faser som löper parallellt. Den fjärde fasen behandlar val av tekniska systemalternativ och den femte fasen behandlar utbyggnad av processmodellerna.



### 6.2.4.1 Fas 4 - val av alternativa tekniska system

I denna fas definieras olika tekniska alternativ varav ett sedan väljs (se Figur 6).



Figur 6 - Val av alternativa tekniska system (Downs, 1992 sid 57)

Utdata från denna fas går till den fysiska designen och består av den valda tekniska lösningen. Lösningen inkluderar en beskrivning av den tekniska miljön, material relaterat till kostnad och vinst, analys av hur det tekniska systemet slår mot organisationen, plan för fortsatt arbetet och hur det tekniska valet matchar kraven på systemet.

#### Steg 410 - definition av alternativa tekniska system

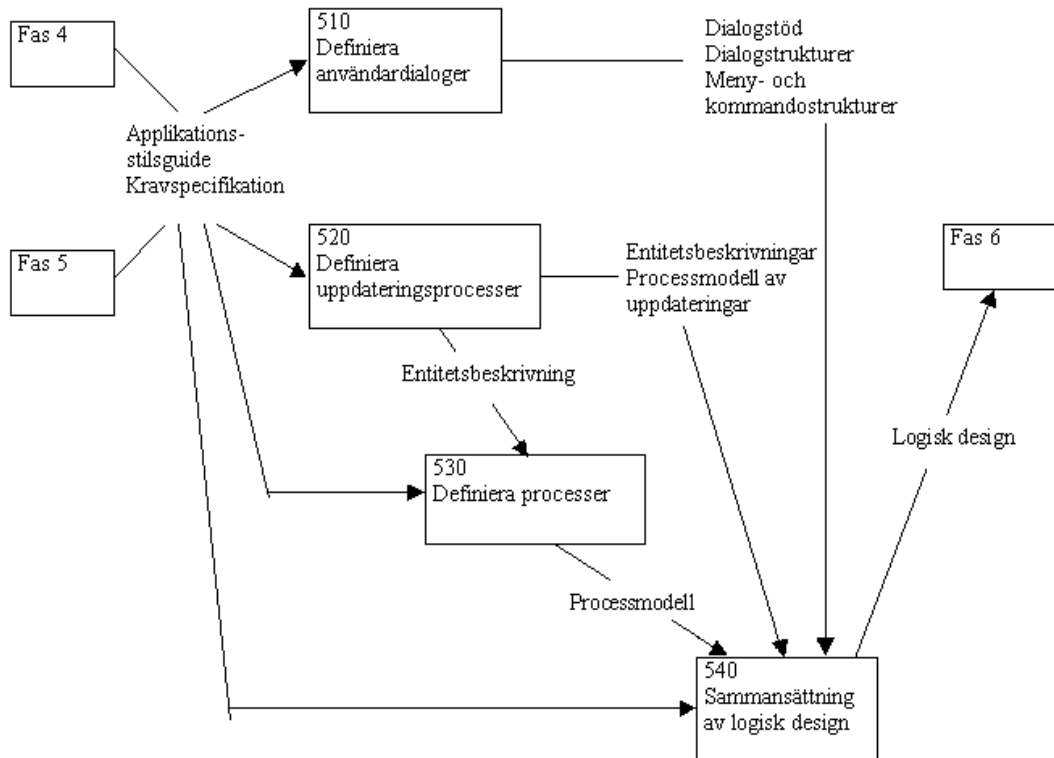
Detta steg innefattar att ta fram ett antal alternativa tekniska system, vilka beskriver möjlig fysisk implementation för att möta de logiska systemkraven. De uppgifter som skall genomföras i detta steg är att först skapa en lista med de begränsningar som måste uppfyllas. Här efter skapas ett antal alternativa lösningar som sedan gallras efter diskussion med användarna. Till kvarstående alternativ skall sedan en beskrivning av den tekniska miljön läggas till samt systembeskrivningen. Slutligen läggs även analyser om hur de tekniska alternativen påverkar kostnad och vinst. De tekniker som används i detta steg är fysisk datadesign, fysisk processdesign och tekniska systemalternativ.

#### Steg 420 - val av tekniskt system

Steg 420 börjar med att systemutvecklarna presenterar de olika alternativen för användarna och ger användarna stöd att utvärdera de olika alternativen. Detta ger möjlighet att diskutera de alternativen med en bredare publik än bara projektledningen. Efter denna diskussion skall de olika alternativen uppdateras för att ta med eventuella synpunkter som framkommit. Efter uppdateringen skall kontroll ske så att alternativen når upp till önskad nivå på systemets tjänster. Slutligen skall en applikationsstilsguide produceras för det nya informationssystemet som baseras på systemets installationsstilsguide. Den teknik som används är tekniska systemalternativ.

### 6.2.4.2 Fas 5 - logisk design

Denna fas utgörs av systemets logiska design. Här designas dialogerna och kravspecifikationen omformuleras till processspecifikationer som i sin tur sedan skrivs ner som programspecifikationer (se Figur 7).



Figur 7 - Logisk design (Downs, 1992 sid 62)

#### Steg 510 - design av användardialogerna

I detta steg fortsätter den logiska designen av dialogerna från fas 3. Det arbete som utförs här baseras på det tidigare arbetet med I/O-strukturerna, funktionerna, menyerna och kommandostrukturerna. Målet med detta steg är att producera logisk dokumentation av dialogerna, vilket görs genom att samla beskrivningar av varje dialogelement som var och en representerar en meningsfull del av dataflödet eller av användarrollerna. För att förenkla designen kan man gruppera dessa delar och skapa logiska grupperingar av dialogelement. Det finns även dialogstrukturdiagram som beskriver den logiska strukturen i dialogen i form av iteration, selektion och sekventiella kommandon. Slutligen görs dialogkontrolltabeller vilka redovisar hur användare förväntas att arbeta genom dialogerna, detta dokumenteras sedan som antingen menyer eller kommandostruktur. Den teknik som används för detta arbete är dialogdesign.

#### Steg 520 - definiera uppdateringsprocesser

Informationen om händelser och funktioner används för att skapa en uppdateringsmodell av processer. Modellen sammanför operationer som bildar och uppdaterar funktioner. Denna bildar ett strukturdiagram tillsammans med operationer som innehåller dess huvuddelar. Produkten i detta steg är ett effektdiagram som visar hur

händelserna interagerar med den logiska databasen. De uppgifter som skall utföras i detta steg är att lägga till tillståndsindikeringar till alla entiteternas livscykler. Vidare skall man uppdatera processtrukturerna till varje händelse från effektdiagrammet och därefter identifiera de operationer som påverkar entiteterna för varje händelse. Dessa operationer skall sedan lokaliseras i processtrukturen och slutligen avgöra vilka handlingsfel som är förknippade till olika händelser. De tekniker som skall användas för att utföra uppgifterna är logisk databasprocessdesign och entitet-händelse-modellering

### **Steg 530 - definiera undersökta processer**

I detta steg skapas en undersökande processmodell. Undersökning består av att titta på en enkel funktion och designa de omgivande processerna. Produkten blir en åtkomstväg som visar interaktionen med den logiska databasen. Vidare definieras också ett strukturschema och associerade operationer. Tekniker för att göra detta är logisk databasprocessdesign.

### **Steg 540 - sammansättning av logisk design**

Det arbete som utförts i denna fas kvalitetssäkras, vilket innebär att kontrollera att allt blivit gjort och att det håller rätt standard. Produkten i detta steg är den logiska design som fås genom allt som skapats i steget. Den logiska designen innehåller:

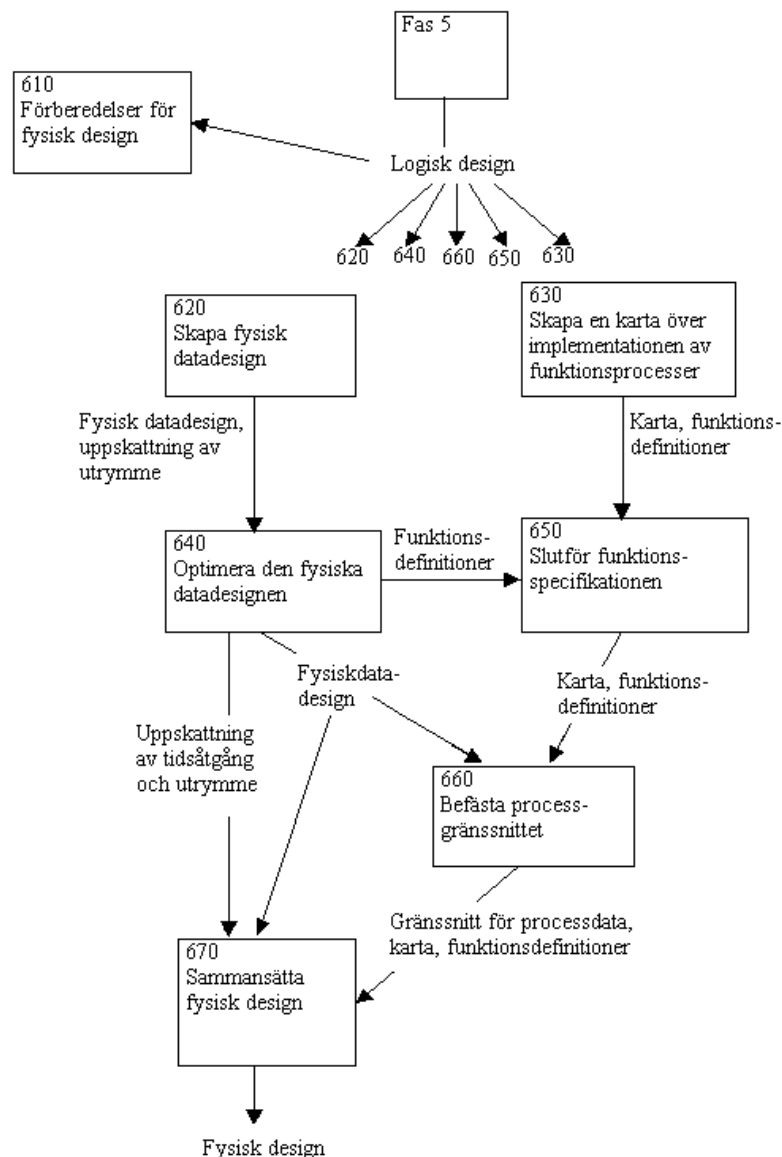
- Logisk processmodell
- Meny och kommandostrukturer
- Kravkatalog
- Datakatalog
- Det krävda systemets logiska datamodell

### **6.2.5 Fysisk design**

I denna modul, som består av en fas, slutförs arbetet och resultatet blir ett färdigt informationssystem. Modulen innehåller endast två, men stora, tekniker; en som behandlar fysisk datadesign och en för den fysiska processspecifikationen. De tidigare faserna har producerat den logiska designen och definitioner om den omgivande miljön i vilken informationssystemet utvecklas och körs. Målet i denna modul är att sammanföra dessa produkter med fysiska designen och resultatet blir ett nytt informationssystem. Själva implementationen av systemet är inte en del av SSADM, men denna sista modul skall underlätta för kommande implementation.

### 6.2.5.1 Fas 6 - fysisk design

Exakt vad som görs i denna fas är beroende av målet i den fysiska omgivningen och vilka verktyg som finns tillgängliga för konstruktion av mjukvara (se Figur 8).



Figur 8 - Fysisk design (Downs, 1992 sid 70)

#### Steg 610 - förberedelser inför fysisk design

Målet för den fysiska miljön är känd från steg 420 där val av teknisk miljö gjorts, dock är det nödvändigt att få fram mer information om denna omgivning. Detaljer om den tekniska miljön, vidare utvecklingsarbete och operationer för det blivande informationssystemet är betecknat som den fysiska omgivningsspecifikationen. Målet för den fysiska miljön klassificeras med hjälp av SSADMs riktlinjer för processsystem, lagring och prestanda på DBMS. Vidare skall standarder för utveckling av mjukvara för hela projektet, produktspecifik mjukvara och namngivning av allt i projektet. Vidare skapas manualer för användarna och utbildning skall påbörjas. Slutligen skall den fysiska designen godkännas av projektledningen.

### **Steg 620 - skapa fysisk datadesign**

Detta steg innehåller två delar. Den första delen innefattar att ta den logiska datamodellen och omvandla den på så sätt att det går att skapa en fysisk design. Syftet är att designen skall minimera åtkomsttiden genom att placera data som sannolikt kommer användas samtidigt i samma fysiska block. Detta görs genom att skapa en fysisk datamodell och ta bort aspekter från den logiska datastrukturen som inte behövs i den fysiska designen. Vidare måste nycklar och icke-nycklar bestämmas samt vilka rotentiteter som den fysiskt relaterade datan skall skapas runt. I den andra delen arbetas med ytterligare transformering, denna gång genom att applicera de regler som är specifika i DBMSen. Fokus ligger på vad som skall grupperas med vad och relationer mellan dessa grupperingar. De uppgifter som skall utföras är att välja blockstorlekar och storlek på varje gruppering så att de passar den valda blockstorleken. Tekniken som används för att utföra arbetet är fysisk datadesign.

### **Steg 630 - skapa en karta över implementation av funktionskomponenter**

Två omfattande delsteg utförs i detta steg. I det första kontrolleras att den logiska beskrivningen av processsystemet är komplett och det andra delsteget bestämmer hur varje komponent i den logiska designen skall byggas. För att utföra dessa två delsteg måste följande uppgifter utföras; minimera dubletter av logiska processer och maximera funktioner eller deras logiska processkomponenter som kan delas av flera processer. Vidare skall syntaxfelsprocesser, navigation och datafel, detaljerad I/O-struktur för sammankoppling med andra system och fysisk dialog så som skärm, navigation och felhantering, designas. För detta steg används tekniken fysisk processspecifikation.

### **Steg 640 - optimering av den fysiska datadesignen**

I detta steg arbetar man på att optimera den lösning som tagits fram, vilket innebär att beräkna det utrymme som krävs för det nya systemet och optimera designen om det är nödvändigt. Vidare skall man även beräkna tidsåtgången för de nya komponenterna i det nya informationssystemet och optimera dessa om det behövs. För att göra detta används tekniken fysisk datadesign.

### **Steg 650 - fullständig funktionsspecifikation**

Uppgifterna är att identifiera logiska processkomponenter inom varje funktion och kombinera dessa komponenter med associerade delar till en programspecifikation, tekniken som används för att göra detta är fysisk processspecifikation.

### **Steg 660 - befästa processdatagränssnittet**

Här länkas den fysiska datadesignen samman med processerna som byggs på en logisk syn på datamodellen. Steget har två delar där den första innefattar att definiera gränssnittet mellan fysiska data. Den andra delen består i att dokumentera validerade rutiner och speciella processmoduler. Syftet är att kontrollera att alla rutiner som länkar data och processer, som alla är delar av gränssnittet, är dokumenterade på ett enda ställe. De uppgifter som skall utföras i detta steg är fastställa om det finns processer som saknar överensstämmelse och hur de är lagrade. Vidare skall nycklar för dessa processer identifieras och vyer av datan skall specificeras. Slutligen skall gränssnittet dokumenteras i kartan över implementation av funktionskomponenter och uppdatera kravkatalogen om det är nödvändigt. Tekniken som används är fysisk processspecifikation.

### **Steg 670 - sammanställa den fysiska designen**

Liksom tidigare moduler avslutas denna modul med kvalitetssäkring av det arbete som utförts i denna fas, vilket innebär att kontrollera att allt blivit gjort och att det håller rätt standard. Vidare kontrolleras även att allt är konsistent och fullständigt. Produkten i detta steg är den fysiska designen som får genom allt som skapats i steget.

### **6.3 Användarmedverkan i SSADM**

Användarmedverkan i SSADM förekommer till största delen i analysfaserna (se fig. 1, sid 35). Metoden ser kommunikation som den främsta egenskapen i användarmedverkan, det vill säga kommunikationen mellan användare och systemutvecklarna, som används som ett verktyg för att systemutvecklarna skall få den information de behöver vid utveckling av det nya systemet. SSADM är ingen metod som till punkt och pricka kan anpassas till användarna, vilket heller inte är något som behöver eftersträvas då metoden skall vara hjälp och riktlinjer för systemutvecklarna. För att ta med användarna i utvecklingsprocessen tar SSADM ofta hjälp från andra ansatser och metoder, exempelvis från sociotekniska ansatser eller metoden COMPACT (se sid. 6), vilken ansats eller metod som väljs är helt beroende på typ av projekt.

SSADM behandlar användarnas intresseområde på olika sätt. För det första finns det i metoden olika strukturerade verktyg, exempelvis dataflödesdiagram och logisk datastruktur, för att förbättra kommunikationen med användarna vid utveckling av modeller för informationssystemet. Detta är till nytta om användarna får lära sig tekniken så att de förstår den och kan använda den. Om så inte är fallet kan det vara frustrerande för användarna och få motsatt effekt, det vill säga att användarna istället för att kommunicera med utvecklarna behåller den information de besitter för sig själva.

För det andra ger SSADM bra stöd för att representera användarnas synpunkter genom att använda tekniken dialogdesign som används för att designa processer och där användarna har möjlighet att påverka designen genom att diskutera processerna med utvecklarna. Här går det dock att ifrågasätta om SSADM verkligen ger stöd för användarna då diskussionen enligt litteraturen oftast består i att systemutvecklarna ställer frågor etc. till användarna, vilket leder till att användarna kan ha svårt att uttrycka hur processen verkligen går till. Detta då det är svårt för användaren att i minnet komma ihåg exakt hur processerna fungerar då detta i det dagliga arbetet oftast går på rutin.

Enligt Waern (1993) finns det en del problem med användarmedverkan i SSADM. Det största problemet är att få fram den information av användarna som behövs för att utveckla systemet. Orsaken till detta är att de uppgifter som användarna utför går på automatik, vilket gör att det är svårt för dem att förmedla denna information genom intervjuer etc. Ett annat problem är att teknikerna är mycket teknikbetingade, vilket kan göra det svårt för användarna att förstå och använda dem. Ytterligare ett problem är att systemutvecklare och användare tolkar saker på olika sätt vilket kan leda till att fel uppstår.

Identifiering av dialoger och design av dialoger börjar mycket tidigt i projektet och har på så sätt positiva effekter på kravkatalogen och processmodellerna. För att öka kommunikationen ytterligare kan specifikationsprototyping användas för att kontrollera kraven och dialogerna. Prototyping kan vara positiv då användarna får ta

del av kraven och dialogerna grafiskt och på så sätt öppet diskutera dessa. Dock får inte användarna själva vara med och utveckla prototypen, vilket görs av systemutvecklarna, utan endast komma med åsikter om en redan färdig prototyp vilket kan ha negativa effekter då användarna på så sätt egentligen inte har något att säga till om. Detta kan leda till irritation bland användarna då de kan tycka att systemutvecklarna har gjort en prototyp som inte alls stämmer med användarnas önskemål. Prototyperna kan vara svåra att förstå, vilket kan få resultatet att användarna inte bryr sig om utvecklingsarbetet. För att ytterligare bygga upp användarnas perspektiv används användardefinierade funktioner som bas för processdesign, vilket innebär att användarna exempelvis blivit intervjuade angående olika funktioner.

Slutligen är det regelbunden inblandning av användarna i utvecklingsprocessen. Vanligtvis finns någon chef över användarna representerad i projektledningen. Likaså finns det även användare representerade med olika kvalifikationer vid de tillfällen då systemutvecklingen utvärderas vid olika tidpunkter. Detta innebär att utvecklarna kontrollerar, med användarnas hjälp, att de förstått olika krav och situationer under utvecklingsprocessen. Detta görs i fas 2, alternativa affärssystem, och fas 4, alternativa tekniska system vid val av olika alternativ.

De sätt på vilket användarna får lyfta fram sitt intresseområde varierar. Det som vanligen används är intervjuer, medverkan i kvalitetssäkring eller inblandning i projektledningen. Om användarna medverkar i projektet är det i form av en användargrupp. Ibland består utvecklingsgruppen nästan bara av användare. Detta är mest lämpat i de tidiga faserna för att analysera krav och specificera dessa men för att göra detta används ofta en mer människo-orienterad ansats, exempelvis metoden COMPACT (se sid. 6).

Användarkommunikation är inte bara koncentrerad på information om informationssystemet utan även på användarnas syn på informationssystemet. En fördel som kommer med användarnas inblandning i utvecklingsprocessen är att de identifierar sig med det nya systemet, på detta sätt blir användarmedverkan en försäljningsstrategi.

### **6.4 Sammanfattning av SSADM**

SSADM är en vattenfallsmetod som mycket liknar traditionella strukturella metoder. Metoden är en datadriven metod då den lägger stor vikt på datamodellering men i den senaste versionen har även användarna fått en mer betydande roll än tidigare. SSADM ger systemutvecklarna mycket detaljerade regler och riktlinjer att arbeta efter. Metoden är mycket strukturerad och standardiserad och erbjuder noggrann dokumentation och är genom detta även lätt att lära ut.

Ramverket för SSADM består av fem moduler, sex faser med tillhörande steg. Detta ramverk kan anpassas på olika sätt utifrån egenskaperna av det informationssystem som skall utvecklas och den organisation där utvecklingen sker. Teknikerna som används i metoden fokuserar mer på tekniska aspekter än på användarmedverkan och används ofta parallellt med varandra för att få en ökad förståelse av situationen.

Det främsta syftet med användarmedverkan i SSADM är kommunikation mellan användare och systemutvecklare. Kommunikation innebär i detta fall att användarna skall förse systemutvecklarna med den information som är nödvändig för att kunna utveckla det nya informationssystemet. Användarna får på så sätt rollen som verktyg till systemutvecklarna, det vill säga användarna ses som hjälp till systemutvecklarna.

## 6 Systemutvecklingsmetoden SSADM

Användarna har i SSADM mindre inflytande än vid mindre traditionella metoder, exempelvis människo-orienterade metoder, och användarmedverkan kan ses som låg. I PD är syftet det omvända, där skall systemutvecklarna vara till hjälp för användarna i systemutvecklingsprocessen. Användarna har mer inflytande och användarmedverkan är hög.

Den användarmedverkan som finns i metoden kan dock betraktas som ett steg mot en ökad uppmärksamhet på användarna. För att finna den information som behövs för att bygga det nya systemet används oftast intervjuer, enkäter, observation, analys av dokument etc. Detta ger dock ingen djup information till systemutvecklarna att använda sig av vid utvecklingsarbetet. Även dataflödesmodellering spelar en stor roll i detta sammanhang, detta då tekniken är enkel och lätt att använda för användarna.



## 7 Integration av PD och SSADM

I detta kapitel undersöks om det är möjligt att integrera PD med SSADM. En integration mellan PD och SSADM är lämpligast att göra under analysfaserna, det vill säga fram till fas 5, logisk design (se Fig. 1 sid 30). Faserna innan handlar till stor del om att hitta information om det gamla och det nya systemet, vilket gör att det är nödvändigt att användarna deltar i processen. Den fysiska designen är mer tekniskt relaterad och kanske svår för användarna att delta i lika stor utsträckning, vilket kanske heller inte är nödvändigt då arbetet med implementering dels är mycket tekniskt relaterat och dels är alla viktiga beslut som rör användarna redan fattade. Integrationen görs genom att titta på de olika faserna i SSADM och undersöka när, hur ofta, på vilket sätt, och med vilket syfte PD kan användas. Nedan följer en beskrivning av hur PD och SSADM skulle kunna integreras i de olika faserna.

### 7.1 Fas 0 - möjligheter

I denna fas är det huvudsakliga syftet att starta upp projektet och avgöra om det alls behövs. De steg som skall genomföras är: definition av problem, val av möjliga alternativ och sammansättning av möjlighetsrapport. Målet är att få reda på vilka resurser som finns, hur problemsituationen ser ut, vilka som skall involveras och utvärdera olika alternativa lösningar på problemet. För att undersöka detta används i SSADM teknikerna dataflödesmodellering, logisk datamodellering, affärssystemval och tekniska systemval.

Här skulle det lämpa sig att använda framtidsseminarium i alla stegen för att kunna diskutera situationen i organisationen där berörda parter tillsammans kommer fram till om projektet skall fortsätta eller inte. Det är inte meningen att använda ett framtidsseminarium i varje steg, utan snarare att genomföra ett framtidsseminarium och använda de olika stegen som stolpar att gå efter i seminariet för att nå fram till målet, det vill säga en rapport med möjligheter. Det finns ingen anledning att ta bort de tekniker som finns i fasen, då dessa är ett bra komplement till framtidsseminariet.

Fördelen med att använda framtidsseminarium är att man kan använda tekniken för att lära ut andra tekniker som kommer att användas i senare faser, exempelvis dataflödesmodellering som är en viktig del i kommunikationen med användarna i SSADM. Dock bör inte tyngden ligga på de övriga teknikerna, utan att använda seminariets tre delar för att ta reda på fasens syfte. Genom att använda framtidsseminarium i denna fas får man, samtidigt som tekniken lärs ut, på tidigt stadium en god bild av problemen i det gamla systemet och önskingar om det nya systemet från användarna och detta material är ett bra underlag att bygga vidare på i kommande faser. Fördelen är även att man tidigt i processen befäster det nya systemet hos användarna, det vill säga användarna får en ägandekänsla, vilket kan underlätta senare arbete och komma ifrån problem med att användarna inte accepterar det nya systemet.

De problem som skulle kunna uppstå är att då tekniken är helt ny för användare kan det ta extra tid att lära ut den. Vidare kan det även ta tid innan användarna känner sig tillräckligt bekanta med tekniken för att använda den till sitt yttersta, det vill säga att användarna skall känna sig säkra och vågar att delta på ett aktivt sätt utan att vara hämmade. Dock är fördelen att när användarna väl lärt sig att använda tekniken behövs ingen mer introduktion då den används vid senare tillfällen, under förutsättning att det

är samma användare som är med under hela processen. Ett problem som kan uppstå är i steget val av möjliga alternativ där syftet är att ta fram kostnader för projektet, förändringar i företaget, risker och livsduglighet i projektet. Orsaken till problemet är att det kan vara svårt att uppskatta exempelvis kostnader då PD används eftersom det är svårt att exakt avgöra hur lång tid projektet kan ta. Ett annat problem i detta steg är att det är för tidigt i utvecklingsarbetet för att de inblandade användarna skall kunna undersöka olika alternativa affärssystem och tekniska system då de ännu inte är så insatta i processen.

### **7.2 Fas 1 - undersökning av nuvarande omgivning**

Under undersökning av nuvarande omgivning går man mer på djupet med det nuvarande systemet, de som skall använda det och deras framtida behov. De steg som skall utföras är: etablera ramverk för analysen, undersöka och definiera krav, undersöka nuvarande processer, undersökning av nuvarande data, härleda en logisk syn på nuvarande tjänster och sammansätta undersökningsresultat. För att få fram fakta om nuvarande data, processer, tjänster och de önskemål som finns används dataflödesmodellering, dialogdesign, logisk datamodellering, kravdefinition och relationsdataanalys.

I det första steget, etablera ramverk för analysen, är det inte nödvändigt att använda PD-tekniker då syftet är att avgöra om projektet skall fortsätta. De övriga stegen har till syfte att definiera krav, processer, data och tjänster, vilket gör att det är av vikt att användarna är med då det är här deras kunskap om verksamheten är avgörande. Eftersom dessa steg sker parallellt är det fördelaktigt att använda framtidsseminarium, för att låta användarna öppet diskutera och komma fram till vilka krav som är nödvändiga, definiera processer och data, liksom härleda nuvarande tjänster. Genom att låta användarna prioritera kraven får man även en indikering om vilka alternativa system som blir aktuella i kommande faser. De tekniker som SSADM använder vid undersökning och definition av krav är dialogdesign och kravdefinition. Vid användning av framtidsseminarium fyller dessa tekniker ingen funktion och är dessutom svåra för användarna att sätta sig in i och kan därför tas bort. Dataflödesmodellering kan dock istället vara en bra teknik att använda tillsammans med framtidsseminarium. Tekniken ger en bra dokumentation över vad man kommit fram till och det borde inte heller vara något större problem att använda den för användarna då den introducerats i föregående fas. Vidare används flödesdiagram i SSADM vid undersökning av processer, vilket även är bra att använda tillsammans med framtidsseminariet då det ger en tydlig bild av flödet av processerna.

Vid undersökning av nuvarande data används i SSADM teknikerna dialogdesign och relationsanalys. Då dessa kan vara komplicerade för användarna att förstå skulle man kunna ta bort dem och ersätta dem med kooperativ prototyping, då teknikernas syfte är att undersöka processer och data. Genom att använda prototyper, för exempelvis entiteter och processerna mellan dessa, får användaren en ökad förståelse för situationen och kan lättare komma fram till vilka processer och entiteter som är nödvändiga. Genom att användaren här tar fram processerna med hjälp av prototyper, blir det även lättare för systemutvecklarna att följa hur användaren tänker och på så sätt förebygga tolkningsfel av informationen. Problem som kan uppstå i detta steg är att den information som tas fram skall vara mycket detaljerad, exempelvis är det här som attribut och nycklar definieras. Denna detaljnivå kan vara svår att uppnå med

framtidseminarium men då kooperativ prototyping används parallellt borde inte detaljerna bli ett problem då prototyper kan vara mycket detaljerade.

I steget härleda en logisk syn på nuvarande tjänster skall den framtagna informationen föras över till en logisk datamodell. Arbetet underlättas genom att systemutvecklare och användare tillsammans tagit fram krav och processer då systemutvecklarna slipper att introducera ännu en teknik, logisk datamodellering, för användarna för att de skall kontrollera att modellen är riktig efter överföringen. Denna överföring kan alltså ske utan att blanda in användarna.

Svårigheter och problem som kan uppstå då PD-teknikerna används i denna fas är att det kan dra ut på tiden då kooperativ prototyping används. Orsaken till detta kan vara att användarna kan vara ovana att arbeta kreativt, inte tidigare har arbetat med tekniken och att de kan ha olika önskemål. Liksom framtidseminarium krävs det att användarna får tid på sig att lära sig tekniken för att kunna plocka fram sin kreativitet, och då de väl lärt sig tekniken går det lätt att använda den i framtiden. Vidare kan problem med olika viljor undvikas genom att användarna får prioritera de krav som kommit fram i framtidseminariet.

### **7.3 Fas 2 - alternativa affärssystem**

I fas 2, som innehåller stegen definition av alternativa affärssystem och val av affärssystem, skall ett affärssystemsalternativ väljas, vilket görs med hjälp av teknikerna dataflödesmodellering, logisk datamodellering och affärssystemalternativ.

I denna fas kan det vara fördelaktigt att använda kooperativ prototyping. Då användarna får vara med och definiera olika alternativ genom prototyping, får de en betydligt bättre bild av vad de olika alternativen innebär för de uppgifter som skall utföras med hjälp av det nya systemet. Det blir enklare att se vad som skiljer de olika alternativen åt samt för- och nackdelar med alternativen. Genom att använda kooperativ prototyping i denna fas skulle det gå att ta bort tekniken affärssystemalternativ i fasen och ersätta den med kooperativ prototyping.

Här skulle man kunna ställa frågan om det verkligen behövs kooperativ prototyping för att uppnå resultat eller om det skulle räcka med alternativa prototyper. Men använder man prototyper som systemutvecklarna utvecklar får inte användarna förståelse om de olika alternativen och har då svårt att avgöra vilket alternativ som är lämpligt att använda till det nya systemet. Resultatet kan bli att man faller tillbaka på de problem som finns med traditionell systemutveckling, exempelvis att inte användarna vill använda systemet. De två andra teknikerna behövs dock vara kvar dels för att användarna nu förstår dataflödesmodellering, under förutsättning att det är samma individer, vilken är bra att använda grafiskt, och dels för att den logiska datamodellen från tidigare faser skall kunna uppdateras av systemutvecklarna. Tidsmässigt är det en fördel att använda kooperativ prototyping då deltagarna i projektet nu är vana att arbeta med tekniken, vilket gör att man kan använda sig av tidigare prototyper då man sätter ihop olika alternativ. Vidare är det överskådligt att använda prototyperna då alternativen skall presenteras och godkännas av projektledningen, vilket förenklar valet av affärssystem.

### **7.4 Fas 3 - definition av krav**

Fas 3 innefattar stegen: definiera behövda systemprocesser, utveckla krävd datamodell, härleda systemfunktioner, utöka krävd datamodell, prototyp av I/O-struktur, utveckla processspecifikation, bekräfta systemets delmål och att slutligen ta fram en kravspecifikation.

Det två första stegen i fasen, definiera behövda systemprocesser och utveckla krävd datamodell, innebär att uppdatera det som framkommit i fas 2. Detta görs enklast genom att använda prototyping tillsammans med ett framtidsseminarium. I framtidsseminariet diskuteras och uppdateras prototyperna, vilket medför att användarnas önskningsar och olika syn på situationen verifieras och kan dokumenteras. Fördelen att använda framtidsseminarium här är att användarna lättare kan förmedla vilka önskemål de har och på så sätt får systemutvecklarna en tydligare bild över de olika definitionerna.

De tekniker som används för dessa steg i SSADM är: dataflödesmodellering, dialogdesign, kravdefinition, logisk datamodellering och relationsdataanalys. PD-teknikerna gör att det går att ta bort dialogdesign, kravdefinition och relationsdataanalys, vilka är svåra för användarna att använda. I de fem sista stegen, härleda systemfunktioner, utöka krävd datamodell, prototyp av I/O-struktur, utveckla processspecifikation och bekräfta systemets delmål är det bra att använda kooperativ prototyping tillsammans med framtidsseminarium. Detta är mycket detaljerat arbete och fördelen med att använda kooperativ prototyping tillsammans med framtidsseminarium är att man får en lättöverskådlig bild av mycket komplicerade delar i systemet, vilket gör att användarna på ett enkelt sätt kan förmedla nödvändig information.

Problem som kan uppstå i dessa steg är att användarna ser på funktioner, processer och I/O-strukturer på olika sätt, vilket kan leda till att konflikter kan uppstå, men då man samtidigt diskuterar de olika funktionerna, processerna och I/O-strukturer i ett framtidsseminarium kan dessa konflikter förebyggas.

Kontentan i fas 3 är att samla den information som fås från de tidigare faserna och framställa en kravspecifikation. I fasen används tekniker som är tekniska och kan vara svåra för användarna att förstå och det kanske heller inte är nödvändigt att de måste förstå dem. Om man i tidigare faser, genom att använda föreslagna tekniker, fått fram den information som är nödvändig för att göra en fullständig kravspecifikation finns det ingen anledning att försöka lära ut dessa komplexa tekniker till dem. Ett alternativ skulle kunna vara att avsluta fasen med ett seminarium, där man kontrollerar att kravspecifikationen täcker det som framkommit i de tidigare faserna.

### **7.5 Fas 4 - val av alternativa tekniska system**

Denna fas liknar fas 2 och kan genomföras på samma sätt, det vill säga genom att göra prototyper på de olika tekniska alternativ som finns för att involverade lättare skall kunna besluta vilket alternativ av tekniskt system som är att föredra.

### ***7.6 Fas 5 - logisk design och fas 6 - fysisk design***

De två sista faserna är mycket tekniskt relaterade liksom de tekniker som används i systemutvecklingsarbetet. Fas 5 innebär att omvandla kravspecifikationen till en logisk design för att sedan fysiskt implementeras i fas 6, fysisk design. Eftersom arbetet präglas helt av tekniska aspekter är det svårt att ta med användarna i denna process, vilket kanske inte heller behövs. Däremot är det i denna fas viktigt att användarna är med och testat och utvärderat systemet, för att kontrollera att det fungerar i enlighet med deras önskningsar.

### ***7.7 Sammanfattning***

Syftet med att integrera PD med SSADM är att få djupare insikt i den information som är nödvändig för att kunna bygga ett informationssystem som motsvarar de önskningsar användarna har på systemet. I utvecklingsprocessen är det lämpligast att använda PD-tekniker i analysfaserna fram till det att kravspecifikationen fastställs. De faser som kommer efter är mycket tekniskt relaterade och kan vara svåra för användarna att greppa och delta i. För att kunna hålla den standard på dokumentation som finns i SSADM är det inte aktuellt att plocka bort dess tekniker helt, men vissa, exempelvis dialogdesign, kan i analysen delvis tas bort och helt ersättas med PD-tekniker, medan andra, exempelvis dataflödesmodellering, är ett bra komplement att använda vid framtidsseminarium och kooperativ prototyping. Generella problem och svårigheter vid integration är dels att det kan dra ut på tiden då användarna måste lära sig att använda teknikerna för att kunna uttrycka sin tankar och önskningsar och dels att användarna har olika önskningsar och tolkningar på olika krav. Detta kan dock undvikas genom att i ett tidigt skede befästa målet med systemutvecklingsprojektet hos användarna så att de arbetar mot ett gemensamt mål. Vidare kan en något längre utvecklingsprocess vägas upp av att det färdiga systemet är bättre anpassat efter användarnas önskemål och på så sätt har högre sannolikhet att implementeras och användas av användarna, det vill säga riskerna som följer med traditionell utveckling minskar betydligt om användarna är med i utvecklingsprocessen från början.

## 8 Genomförande av intervju

Genomförandet av intervjun utfördes som beskrivet i planen för arbetet (se kap. 4.3). Dock genomfördes endast en intervju istället för tre till fyra intervjuer. Orsaken var svårigheten att få tag i intervjupersoner med de utsatta kriterierna. Intervjun var en direktintervju som förrättades på det företag där intervjupersonen arbetade. Intervjun tog cirka 40 minuter. Intervjun inleddes med en presentation av arbetet och syftet med arbetet. En ingående beskrivning hur PD-teknikerna framtidsseminarium och kooperativ prototyping användes och vad teknikernas syfte är gjordes för att intervjupersonen skulle ha bakgrunden och kunskapen att besvara fråga tio. De planerade frågorna ställdes i den ordning som var bestämt, men till dessa följde även en del följdfrågor för att förtydliga de frågor som fanns och för att få djupare förståelse i vad intervjupersonen berättade om. Till fråga 5 (ger metoden stöd för användarmedverkan) ställdes följdfrågan: Vad gör man i projekteringen? Denna fråga ställdes då det var oklart vad en av faserna i metoden innebar, vilket syfte fasen hade.

I fråga 9, där de tekniker som användes för att få information av användarna behandlas, ställdes följdfrågorna: Lär ni ut datamodellering innan tekniken används? Har detta fungerat bra, använder ni inte intervjuer eller enkäter alls? Ritar användarna prototyperna själva vid prototyping? Följdfrågorna till fråga 9 ställdes för att gå djupare på de tekniker som användes och hur de användes. Detta för att kunna dra paralleller till det sätt på vilket teknikerna används i SSADM.

Fråga 10 (anser ni att det sätt att representera användarna i er metod är bra eller skulle det kunna göras på något annat sätt, i så fall hur) fick följande följdfrågor: Har ni hört något ifrån användarna när ni använt den traditionella metoden att de velat vara med mer? Hur många var det i projektet? Hur många användare var med? Var det ”riktiga” användare eller ansvariga? Efter fråga 10 var svaret mycket otydligt och eftersom jag ville få fram mer konkret om verkligen användarna representeras på ett bra sätt, ställde jag följdfrågorna i syfte att få reda på om man av antalet användare kunde dra slutsatsen om de var väl representerade och även om de själva velat vara mer representerade i projekten.

Själva intervjufrågorna var avslutade gjordes på cirka 20 minuter. För att kontrollera att jag inte missat något eller om intervjupersonen hade mer att tillägga till diskussionen efter de planerade frågorna var färdiga frågades även följande: Är det något mer som du har tänkt på nu som kan tilläggas? Vilka användare var med i detta större projekt? Vad fick ni för respons på det systemet? Efter den första följdfrågan ställts berättade intervjupersonen om ett annat större projekt, än det mindre som refererades till tidigare, härav kom de andra följdfrågorna för att se om det fanns någon skillnad mellan det större och det mindre angående användarmedverkan.

Efter det att frågorna var avslutade gjordes en grundligare genomgång av metoden som fanns dokumenterad i pärmar. Genom detta fick jag en djupare insikt i metodens olika faser och kunde se vad som stod i varje fas när det gällde användare och användarmedverkan och fick även möjlighet att anteckna detta. Genomgången var mycket positiv, då det efter att ha sett metodens uppbyggnad är betydligt lättare att jämföra med SSADM.

Den information som framkom under intervjun var användbar för att dra slutsatser om integrationen mellan traditionella metoder och PD-teknikerna. Svaren på frågorna stämde väl överens med de svar som var väntade efter att ha studerat SSADM, det vill

## 8 Genomförande av intervju

säga syftet med användarmedverkan, hur metoden var uppbyggd etc. Det var dock svårare att hålla intervjun så ostrukturerad och öppen som var avsikten från början. Även om frågorna var enkla och tydliga krävdes ändå ledande följdfrågor för att få igång en öppen diskussion och få fram svar på det som efterfrågades.

## 9 Resultat från intervjun

Intervjupersonen har arbetat med systemutveckling sedan 1986 och har under denna tid haft olika roller. När företaget arbetade med den traditionella metoden AU, administrativ utveckling, fanns det inte så många olika roller, men var antingen utvecklare eller systemutvecklare. Då projekten var ganska små delades rollerna inte upp i mindre bitar utan alla deltagare arbetade tillsammans under hela utvecklingsprocessen.

Metoden intervjupersonen använt är AU och kategoriseras som traditionell. Metoden består av tre faser: förstudie, projektering och genomförande. De tekniker som främst används är datamodellering och dataflödesmodellering. Det senaste projektet med AU ägde rum 1992, 1993 och då var även projektledarna med och utvecklade.

Förstudien innebär att beskriva nuläget, analysera problem, formulera förändringsförslag och utforma ett grovt lösningsförslag. Projekteringen innebär att man går djupare in på detaljer om verksamheten och lösningen till problemen. Man analyserar verksamheten, utformar lösningar, anpassar organisationen och utformar arbetsrutiner. Syftet med projekteringen är att analysera verksamheten och specificera en lösning. I genomförandet detaljutformas lösningen, vilken sedan realiseras och implementeras. I metoden ses användarmedverkan som en policy, men det är inte specificerat i metoden om hur och när det skall finnas användarmedverkan.

Vid systemutvecklingen har användarna varit inblandade i förstudien och projekteringen. I genomförandefasen har de varit med i slutet för att testa det nya systemet.

Användarnas roll i systemutvecklingen är att i förstudien vara med och starta upp projektet och till viss del driva denna del av utvecklingen. Då själva metoden börjar användas, i projekteringen, är det systemutvecklarna som driver processen medan användarna är med och talar om hur saker och ting fungerar. Syftet med användarmedverkan i metoden är att få fram information av användarna då det är de som vet hur det fungerar i verksamheten och sitter på den information som behövs.

För att få den information som användarna besitter används dataflödesdiagram, vilket är ett bra hjälpmedel för användare och systemutvecklare att förstå varandras språk. Tekniken lärs ut till användarna i en kort genomgång innan den skall användas och under utvecklingen används sedan tekniken tillsammans och förklaras mer under tidens gång, vilket har fungerat bra. I metoden används inte tillvägagångssätt som intervjuer eller enkäter, dock frågar man användarna om detaljer, exempelvis om hur de vill ha bilder etc. Efter detta så tas olika alternativ på bilder fram av systemutvecklarna, så att användarna sedan får kommentera dessa.

Vid de mindre projekten har gruppen bestått av cirka tio personer, varav tre har varit användare som skall arbeta med det nya systemet. I större system som utvecklats är situationen lite annorlunda. I de projekten används representanter för användarna exempelvis en kostnadsansvarig. Är man många i projektet är det svårt att ha många användare med, eftersom alla har olika intressen och det blir då svårare att driva processen. Nackdelen är dock att vissa anpassningar måste göras för att kunna tillgodose alla kraven och att det ibland måste göras olika varianter på saker för att passa. Vid förändringarna och anpassningarna diskuteras detta med representanterna som var med under tiden. Intervjupersonen anser att det alltid går att representera användarna bättre då det är mycket viktigt att dessa är med, men det sätt de har arbetat



## 9 Resultat från intervjun

på har fungerat bra och har inte fått anmärkningar från användarna att det skulle kunna ha gjorts bättre.

Slutligen ansåg intervjupersonen att det var möjligt att integrera PD-teknikerna framtidsseminarium och kooperativ prototyping med en traditionell systemutvecklingsansats. Vidare lade intervjupersonen till att en integration skulle kräva mycket engagemang från både användare och systemutvecklare samt att de båda parterna var tvungna att anpassa sig efter varandra, det vill säga att användarna måste förstå hur systemutvecklare arbetar och systemutvecklare måste sätta sig in i verksamheten och i användarnas arbete.

Vid jämförelse av SSADM och AU ser man att metoderna är mycket lika, även om de innehåller olika faser. Förstudien är den samma som SSADMs förstudie och liksom SSADM tillhör inte förstudien själva metoden, det vill säga det går att utföra förstudien på olika sätt. Projekteringen är den samma som SSADMs faser fram till fas fem, logisk design. Fasen genomförande i AU motsvaras av logisk design och fysik design i SSADM. De två metoderna liknar även varandra i synen på användarna. Detta innebär att användarna i båda metoderna har till syfte att stödja systemutvecklarna med information om verksamheten och det gamla systemet. Vidare deltar användarna till största delen i de tidiga faserna i utvecklingsprocessen, men även i slutet av utvecklingen genom tester av systemet.

Under intervjun betonar intervjupersonen återkommande vikten av att användarna är med under systemutvecklingsprocessen. Angående de större projekten kunde detta dock uppfattas som ett försök att få det att verka som användarna är representerade på ett bra sätt, fast de kanske egentligen inte är med så mycket i själva processen. Likaså påpekade intervjupersonen att allt fungerade bra under systemutvecklingsprocessen, men då följdfrågor som var mer preciserade ställdes framkom att det fanns problem som orsakades av att användarna inte direkt var med i utvecklingen utan istället representerades av användarrepresentanter. Kontentan skulle kunna vara att intervjupersonen ville att det skulle låta som metoden AU var mer anpassad till användaren än vad den i verkligheten var. Detta kan bero på att systemutvecklare börjar bli mer medvetna om användarinvolvering och att det har positiva effekter på systemutvecklingsprojekt och för att det skall se bra ut utåt om de är engagerade i användar-medverkan.

## 10 Slutsats

Syftet med arbetet är att undersöka om det går att integrera PD-tekniker med den traditionella systemutvecklingsansatsen.

Från litteraturstudien har det framkommit att användarmedverkan förekommer i form av dels informationsinsamlande, exempelvis intervjuer, enkäter, etc. och dels i form av tester och utvärdering av det nya systemet. I metoden är användarna till största delen involverade i analys- och designfaserna, medan de står utanför systemutvecklingsprocessen i implementationsfaserna. Vid integration mellan PD-teknikerna framtidsseminarium och kooperativ prototyping och SSADM är det möjligt att ersätta vissa tekniker i SSADM med PD-tekniker, samt komplettera SSADMs tekniker med PD-tekniker. Vidare är det vid integrationen mest lämpat att använda PD-teknikerna under analysfaserna, då syftet är att samla information om verksamheten och det gamla informationssystemet. De senare faserna i metoden är mer tekniskt relaterade vilket gör det svårt att involvera användarna på ett effektivt sätt.

I intervjun framkom att den traditionella metoden som användes, var uppbyggd på samma sätt som SSADM. Dels i form av uppbyggnad av själva metoden, det vill säga faser, tekniker etc. och dels i synen på användarna. I AU deltar användarna i form av intervjuer och är till stöd för systemutvecklarna vid utformning av det nya informationssystemet.

Genom att studera resultaten från litteraturstudien och intervjun kan slutsatsen dras att användarmedverkan förekommer i de traditionella systemutvecklingsmetoderna under analys- och designfaserna. Av denna orsak är det mest lämpat att använda PD-teknikerna framtidsseminarium och kooperativ prototyping i dessa faser. Dock innebär inte detta att det generellt går att använda alla olika PD-tekniker under dessa faser, eftersom teknikerna har olika syften (se kap. 2.5). Då alla traditionella systemutvecklingsmetoder är uppbyggda på liknande sätt är det helt möjligt att använda PD-teknikerna framtidsseminarium och kooperativ prototyping i faserna fram till att kravspecifikationen utformas. Detta genom att dels ta bort vissa tekniker och helt ersätta dem med PD-teknikerna och dels genom att använda PD-teknikerna som ett komplement till de tekniker som redan finns, exempelvis dataflödesmodellering.

För att en integration mellan de båda ansatserna skall vara möjlig krävs det ett engagemang från både användare och systemutvecklare, det vill säga att båda parterna är villiga att lyssna och lära ifrån varandra. Det kräver att de inblandade talar samma språk, det vill säga förstår varandras sätt att uttrycka sig, exempelvis genom facktermer.

## 11 Diskussion

Detta kapitel innefattar en diskussion runt arbetet. Det innehåller en utvärdering av arbetet samt en beskrivning av de erfarenheter som fåtts under arbetets gång.

### 11.1 Utvärdering av arbetet

För att kunna besvara arbetets frågeställningar började jag med att studera traditionella systemutvecklingsmetoder, hur de är uppbyggda, vilka fördelar och nackdelar som finns och hur användarna är representerade i systemutvecklingsarbetet. Inom detta område finns det väl dokumenterad information, men då det gällde hur användarna var representerade fanns det inte lika tydliga beskrivningar. Av denna orsak har jag fått dra egna slutsatser om när och hur användarna representeras i systemutvecklingen. Dessa slutsatser fick jag sedan bekräftade under intervjun där användarna representerades på samma sätt i verkligheten som enligt litteraturen och de slutsatser som dragits.

Efter studier av den traditionella ansatsen studerade jag de för frågeställningen aktuella PD-teknikerna. Även inom detta område fanns det gott om litteratur i form av artiklar och böcker att använda. Det fanns dock mycket lite information att hämta som behandlade dessa ansatser tillsammans och genom detta har jag fått dra egna slutsatser om hur en integration skulle kunna genomföras. Det finns många olika sätt som denna integration skulle kunna utföras på, men för att finna det bästa skulle man behövt testa det i verkligheten och utvärdera konsekvenserna av olika alternativa integrationer, vilket i detta arbete skulle vara svårt på grund av tidsbrist.

I kapitel 6 har en beskrivning av systemutvecklingsmetoden SSADM gjorts, dess uppbyggnad, tekniker och användarmedverkan. I detta kapitel har jag försökt att sätta mig in i hur metoden fungerar och hur man arbetar med metoden. Detta var dock svårt utan praktisk erfarenhet av metoden, då den innehåller många och väl definierade steg. Det var mycket information att få grepp om genom att bara läsa sig till den, många steg liknade varandra och det var lätt att blanda ihop begrepp och svårt att få struktur på den.

Vidare har jag även saknat praktisk erfarenhet av systemutvecklingsprojekt, vilket gör det svårt att dra slutsatser om vilka resurser som finns till förfogande vid ett systemutvecklingsprojekt och hur projektet påverkas av en integration mellan två olika ansatser och hur denna integration skulle kunna göras på bästa sätt. Jag har endast utgått ifrån en teoretisk bakgrund, vilken kanske inte alltid stämmer överens med verkligheten. Dock har en del av teorin bekräftats av den intervju som gjordes med en systemutvecklare.

Eftersom endast en intervju har utförts går det att ifrågasätta generaliserbarheten av de traditionella systemutvecklingsmetoderna. I arbetet har SSADM tillsammans med den metod som behandlats i intervjun legat till grund för hur alla traditionella systemutvecklingsmetoder är uppbyggda och hur de olika metoderna behandlar användarmedverkan. Detta ger en relativt smal bild av traditionella systemutvecklingsmetoder då det finns många metoder inom ramen för den traditionella systemutvecklingsansatsen. Det skulle ha varit fördelaktigt att haft ytterligare några intervjuer för att kunna stärka denna generalisering vid integrationen mellan PD-teknikerna och den traditionella ansatsen, och på så sätt ytterligare bekräfta att det är möjligt att integrera PD-tekniker och den traditionella systemutvecklingsansatsen.

## ***11.2 Kritisk granskning av litteraturen***

Den litteratur jag använt mig av har främst behandlat traditionell systemutveckling, SSADM och PD.

Vid beskrivning av systemutveckling i bakgrunden har jag främst använt mig av artiklar för att få så aktuell information som möjligt, detta då framsteg på området ändras mycket fort. Vid beskrivning av PD har jag dels använt mig av artiklar och dels böcker. Eftersom det finns relativt lite nyskrivet litteratur om denna ansats är böckerna jag använt lite äldre men dock väl refererade till olika författare i de artiklar jag hittat inom området. I kapitlet som behandlar PD-tekniker har jag använt mig av de böcker jag använt till bakgrunden. Detta då dessa böcker är mycket ansedda inom PD och de böcker som alltid finns refererade till i artiklar.

Då jag beskrivit den traditionella systemutvecklingsansatsen har jag till största delen använt två olika författare som båda är ansedda inom detta område. Jag har valt två olika författare för att kunna jämföra deras beskrivningar på den traditionella ansatsen och deras åsikter om den.

Vid beskrivningen av systemutvecklingsmetoden SSADM har jag använt två olika böcker. Den första, Downs (1992), behandlar hela metoden SSADM och den andra, Meldrum (1993), behandlar teknikerna i SSADM. Båda dessa böcker är skrivna av författare som har mycket erfarenhet av systemutveckling och som är ansedda inom området. Av denna orsak har jag gjort bedömningen att det skulle räcka med att använda dessa två böcker.

## ***11.3 Erfarenheter under arbetets gång***

Efter att ha funderat över vilka olika metoder och tillvägagångssätt att genomföra mitt arbete på kom jag fram till att använda mig av en litteraturstudie med stöd från intervjuer (se kap 4). Av litteraturstudien har jag lärt mig att kritiskt granska olika källor och hämta in information som varit aktuell för just min problemställning och att väga dessa mot varandra.

En svårighet som jag stött på med litteraturstudien är att det tar mycket lång tid att sätta sig in i informationen och hitta rätt i olika artiklar och böcker. Vidare har det varit svårt att sätta sig in i exempelvis SSADM genom att bara studera litteraturen. Det ger inte alls samma insyn i metoden som om jag haft praktisk erfarenhet av systemutveckling. Vidare var det även svårt att genomföra en integration mellan de båda ansatserna då jag saknar praktisk erfarenhet av såväl systemutvecklingsprojekt i traditionell systemutveckling som systemutveckling med PD.

Ett annat hinder jag stötte på var att det var svårt att hitta lämpliga personer att intervjua om användarmedverkan i traditionell systemutveckling. Orsaken till detta är att det finns mycket få företag som arbetar med hela systemutvecklingsprocessen i Skövde och runt omkring. De flesta köper in färdiga system från de större företagen i exempelvis Stockholm.

## ***11.4 Resultat***

PD är en systemutvecklingsansats som tagits fram i Skandinavien, vilket kan påverka användningen av det tillvägagångssätt man arbetar på. Då PD exempelvis använts i

USA har det visat sig att resultatet inte varit lika tillfredsställande, som då man använt det i Skandinavien. Orsaken till detta är att organisationerna ser olika ut i olika kulturer, exempelvis arbetar företag i USA efter en mer hierarkisk organisation än vad företag i Skandinavien gör (Ehn, 1993). Av denna anledning skulle man kunna dra slutsatsen att olika företagskulturer påverkar användningen av PD. För att PD skall kunna användas bör organisationen vara relativt platt.

En annan aspekt som kan påverka användandet av PD inom företaget är demokratin på arbetsplatsen, det vill säga vilken rätt de anställda har att påverka i olika situationer (Ehn, 1993). I Skandinavien har företagen en hög demokratisering på arbetsplatsen, liksom en stark facklig organisation, detta jämnar ut skillnaderna mellan anställda samt skapar bättre relationer mellan anställda och arbetsgivare, vilket är en förutsättning för att PD skall kunna fungera effektivt. I andra länder där denna demokratisering inte finns kan det vara svårare att arbeta med PD då klyftan mellan arbetare och arbetsgivare ofta är större. Det kan då vara svårt för de båda parterna att lyssna till varandra och på så sätt samarbeta mot ett gemensamt mål.

Vidare går det även att diskutera om de traditionella systemutvecklingsmetoderna är på väg bort från marknaden. Min uppfattning är att man allt mer idag ser att de objektorienterade ansatserna används mer än de traditionella ansatserna, men även detta kan skilja sig mellan olika länder. I exempelvis Storbritannien, där SSADM är utvecklad, är den traditionella ansatsen mer djupt rotad medan den objektorienterade ansatsen används mycket utbrett i Sverige. Oavsett vilken ansats man arbetar efter är det viktigt att användarna är med i systemutvecklingsprocessen och härav är det nödvändigt att integrera PD med olika ansatser i största möjliga utsträckning. Vidare är det även viktigt att försöka anpassa PD till olika kulturer och politiska klimat, framförallt i de organisationer som är hierarkiskt uppbyggda och har en låg demokratisering, då det i de verksamheterna kanske är svårast att förankra ett nytt informationssystem hos användarna. Detta då de som skall använda det nya informationssystemet kan befinna sig på en lägre nivå i hierarkin och kanske inte har så stort inflytande över de beslut som fattas angående införandet av ett nytt informationssystem som de sitter på en högre nivå i hierarkin.

En annan faktor att ta hänsyn till vid användning av PD är hur det påverkar användarna. En förutsättning för att det skall fungera tillfredsställande är dels att de anställda verkligen vill och är intresserade av att delta i systemutvecklingsprocessen och dels att de får tid till att delta i projektet. Finns det inget intresse bland användarna blir resultatet av utvecklingsarbetet kanske inte bättre bara för att PD används, snarare tvärt om. Är användarna oengagerade är de kanske inte villiga att delge sina kunskaper och den information de besitter, vilket får konsekvensen att informationssystemet blir oanvändbart. Får användarna inte heller det tidsutrymme som behövs för att delta i processen kan de inte på ett aktivt sätt bli involverade i systemutvecklingsprocessen, vilket får till följd att projektet kan misslyckas.

### ***11.5 Förslag till fortsatt arbete***

Det finns många aspekter som skulle kunna arbetas vidare på efter detta arbete, men det mest intressanta skulle vara att prova integrationen på ett verkligt fall, det vill säga använda sig av en traditionell systemutvecklingsmetod integrerad med PD-tekniker i ett riktigt systemutvecklingsprojekt. Genom att göra detta skulle man även kunna se

vad som skulle kunna göras på ett bättre sätt och även undersöka om det går att använda andra PD-tekniker än framtidsseminarium och kooperativ prototyping.

Ett annat intressant fortsatt arbete skulle vara att utforma en ny systemutvecklingsmetod där man använder PD tillsammans med traditionell systemutveckling och tar fram faser, steg, tekniker etc. Genom att utforma en ny metod skulle det gå att ta tillvara på den traditionella ansatsens positiva egenskaper tillsammans med det fördelarna som kommer med PD. På detta sätt skulle man få ett strukturerat sätt att tillämpa PD på samt komma ifrån problem som följer med PD, exempelvis att PD ofta kan dra ut på tiden. Vid utarbetande av en ny metod skulle man även få den erfarenhet inom området som idag saknas genom att vid upprepade tillfällen använda metoden i systemutvecklingsprojekt.

Ännu en aspekt att arbeta vidare på är de ekonomiska faktorer som följer användandet av PD. För att tillvägagångssättet skall få genomslagskraft är det nödvändigt att kunna beräkna eventuella vinster och förluster som kommer med ett införande av PD. Om det inte är möjligt att beräkna de ekonomiska konsekvenserna av tillvägagångssättet kan det vara svårt att få företag att vilja använda PD och befästa fördelarna med det.

En viktig aspekt inom området är användarna och vad de har för attityd till PD. Detta är ett ännu ett förslag till fortsatt arbete, det vill säga att undersöka hur användarna ser på användandet av PD och hur de påverkas av processen. Genom att undersöka detta kunde man få reda på användarnas intresse för PD och hur man skulle kunna anpassa PD för att på ett mer effektivt sätt ta tillvara på användarnas kompetens.

Ytterligare ett förslag på fortsatt arbete är att integrera PD med andra metoder än de traditionella systemutvecklingsmetoderna, exempelvis inom den objektorienterade ansatsen då denna ansats blir allt mer vanlig att använda. Likaså skulle det vara intressant att undersöka hur olika kulturella aspekter påverkar PD, på vilka sätt PD skall anpassas till kulturella aspekter och olika politiska klimat.

## Referenser

- Andersen, E. S. (1994) *Systemutveckling – principer, metoder och tekniker*. Studentlitteratur, Lund.
- Avison, D. E. och Fitzgerald G. (1994) *Information Systems Development: Methodologies, Techniques and Tools*. Alfred Waller, Oxfordshire.
- Avison, D. E. och Fitzgerald G. (1998) *Information Systems Development: Methodologies, Techniques and Tools*. McGraw-Hill.
- Bravo E. (1993) The Hazards of Leaving out the Users. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Bødker, S., Greenbaum, J., och Kyng, M. (1991) Setting the stage for Design as Action. I: Greenbaum och Kyng (red.), *Design at work: Cooperative Design of Computer Systems*. Lawrence Erlbaum Associates, Hillsdale, New Jersey.
- Bødker, S., Grønbaek, K., och Kyng, M. (1993) Cooperative Design: Techniques and Experiences From the Scandinavian Scene. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Cherry, C., och Macredie, R.D., (1999) The Importance of Context in Information System Design: An Assessment of Participatory Design. *Requirements Engineering*, 4: 103-114.
- Dahlström, K., (1996) *Från datainsamling till rapport- att göra en statistisk undersökning*. Studentlitteratur, Lund.
- Dawson, C., (2000) *The essence of Computing Projects: a Student's Guide*. Prentice Hall, Essex.
- Ehn, P. (1993) Scandinavian Design: On Participation and skill. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Ehn, P., och Sjögren, D. (1991) From Systems Descriptions to Scripts for Action. I: Greenbaum och Kyng (red.), *Design at work: Cooperative Design of Computer Systems*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Ehn, P., och Kyng, M. (1991) Cardboard Computers: Mocking-it-up or Hands-on the Future. I: Greenbaum och Kyng (red.), *Design at work: Cooperative Design of Computer Systems*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Friedman, A. (1989) *Computer System Development: History, Organization and Implementation*. Galliard, Great Yarmouth.
- Flensburg, P., och Friis, S. (1999) *Mänskligare datasystem-utveckling: användning och principer*. Studentlitteratur, Lund.
- Greenbaum, J. (1993) A Design of One's Own: Towards participatory Design in the United States. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.

- Grudin, J. (1993) Obstacles to Participatory Design in Large Product Development Organizations. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Grønbaek, K., Grudin, J., Bødker, S. och Bannon, L. (1993) Achieving Cooperative System Design: Shifting From a Product to a Process Focus. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Holtzblatt, K., och Jones, S. (1993) Contextual Inquiry: A Participatory Technique for System Design. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Kensing, F., och Madsen, H. M. (1991) Future Workshops and Metaphorical Design. I: Greenbaum och Kyng (red.), *Design at work: Cooperative Design of Computer Systems*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Klöckner, K., Pankoke-Babatz, U., och Prinz, W. (1999) *Experiences with a cooperative design process in developing a telecooperation system for collaborative document production*. Behaviour & information technology, 1999, vol.18, no.5, 373-383.
- Waern, Y. (1993) Från människa till datoranvändare I: Lennerlöf (red.), *Människor-Datateknik-Arbetsliv*, Publica.
- Mumford, E. (1993) The Participation of Users in Systems Design: An Account of the Origin, Evolution, and Use of the ETHICS Method. I: Schuler och Namioka (red.), *Participatory Design: Principles and Practices*. Lawrence Erlbaum. Hillsdale, New Jersey.
- Nurminen, M. I. (1988) People or Computers: Three Ways of Looking at Information Systems. Studentlitteratur, Lund.
- Patel, R. och Davidson, B. (1994) *Forskningsmetodikens grunder*. Studentlitteratur, Lund.
- Rosengren, P. och Wingstedt, U. (1993) *En metod för användbarhetsdriven prototyputveckling*. SISU dokument, nr 13.
- Schuler, D. och Namioka, A. (1993) *Participatory Design: Principles and Practices*. Lawrence Erlbaum, Hillsdale, New Jersey.
- Ziya Aktas, A. (1987) *Structured Analysis & Design of Information Systems*. Prentice-Hall, Englewood Cliffs, New Jersey.



## Bilaga 1 - Intervjufrågor

1. Hur länge har ni arbetat med systemutveckling?
2. Vilken eller vilken roll/roller har ni haft i systemutvecklingsarbetet?
3. Vilka metoder har ni använt vid systemutveckling och hur är dessa klassificerade ( traditionell, objektorienterad etc)? *Bestäm därefter vilken metod ni ska utgå från i resten av frågorna.*
4. Hur är metoden uppbyggd, faser, tekniker etc?
5. Ger metoden stöd för användarmedverkan?
6. Vilken är användarnas roll vid systemutvecklingen med er metod?
7. När i systemutvecklingsprocessen deltar användarna och på vilket sätt?
8. Vad är syftet med användarmedverkan för er i systemutveckling?
9. Vilka tekniker används för att få information ifrån användarna?
10. Anser ni att det sätt att representera användarna i er metod är bra eller skulle det kunna göras på något annat sätt, i så fall hur?
11. Tror ni att PD kan användas inom traditionell systemutveckling? Varför/varför inte?

## Bilaga 2 – Presentation av författare

De författare jag kommer att använda till litteraturstudien är följande:

*Mary Meldrum* vid University of Sunderland. Författaren är en senior föreläsare med många års erfarenhet av undervisning i analys och design. Vidare undervisar hon även dataspecialister i att använda SSADM.

*John S. Hares* Information Technology Associates, UK. Författaren är erfaren konsult och användare av SSADM.

*Ed Downs* har erfarenhet inom kvalitetssäkring, *Peter Clare* har arbetat i många systemutvecklingsprojekt, en strategisk studie och förvaltat ett informationscenter. Båda författarna föreläser nu vid Lancashire Polytechnic, UK. Den tredje författaren är *Ian Coe* och är senior konsult vid Paxus Services Ltd, Nya Zeeland.

*Joan Greenbaum* är professor vid Computer Information Systems i LaGuardia Community College, City University of New York. Många av projekten hon medverkat i behandlar effekterna av tekniska förändringar på kvinnliga anställda. Hon har varit involverad i utvecklingen av aktiva inlärningsstrategier för att hjälpa människor att lära sig om teknik och ekonomi för att kunna hantera dessa områden mer effektivt på arbetsplatsen.

*Morten Kyng* är assisterande professor vid Computer Science Department i Aarhus University, Danmark. Han undervisar i systemutveckling och forskar om tekniker och verktyg för kooperativ design av datasystem. Han har gjort betydande arbete om hur man ökar slutanvändarnas influens på designen av datasystem.

*Kim Halskov Madsen* är assisterande professor vid Institute of Information and Media Science, Aarhus University, Danmark. Han undervisar i programutveckling och forskar om tekniker och verktyg för kooperativ design.

*Finn Kensing*, är assisterande professor vid Computer Science Department, Roskilde University Center, Danmark. Han forskar om teorier och metoder för systemutveckling, inom vilket område han också undervisar.

*Susanne Bødker*, är assisterande professor vid Computer Science Department, Aarhus University. Hennes forskning fokuserar på mänsklig användning av datorteknik och designprocessen kopplat till detta område. Hon undervisar i systemutveckling och MDI.

*Kaj Grønbaek*, är assisterande professor vid Computer Science Department, Aarhus University. Han undervisar i systemutveckling och forskar om tekniker och verktyg för kooperativ design av datorsystem. Speciellt har han behandlat studier om inblandning av slutanvändaren vid systemutveckling och hur detta kan underlättas med olika tekniker och verktyg för kooperativ prototyping.

Av de tre böcker som behandlar systemutvecklings metoden SSADM, Downs, Meldrum och Hares, har jag valt att använda Meldrum och Downs detta då litteraturen är välkända böcker som flera andra författare har refererat till. Den tredje boken om

SSADM behandlar mer användandet av SSADM inom databashantering och mycket lite om metoden i sig, dess faser och tekniker. De övriga författarna finns i böckerna Design at Work av Greenbaum och Kyng och Participatory Design: Principles and Practices av Schuler och Namioka. Båda dessa böcker är kända och mycket refererade inom PD och även de bland de fåtal böcker som behandlar PD.