

# **Standardsystemsutveckling**

## **– Konsten att göra det specifika generellt**

Magisteruppsats, 10 poäng, inom Systemvetenskaplig programmet

*Framlagd:* Januari, 2005

*Författare:* Henrik Andersson

Johan Friberg

*Handledare:* Jonas Hedman

LUNDS UNIVERSITET

Informatik

# Standardsystemsutveckling

## – Konsten att göra det specifika generellt

© Henrik Andersson  
Johan Friberg

Magisteruppsats framlagd januari, 2005

Omfång: 60 sidor

Handledare: Jonas Hedman

### Resumé

*Denna uppsats redogör för hur standardsystem utvecklas i utvalda, undersökta miljöer. Syftet med denna uppsats är att delge kunskap om utveckling av standardsystem samt att identifiera och belysa de steg som utvecklingsprocessen innefattar. Studien är genomförd på två fallföretag som tillverkar standardsystem och har använt kvalitativa intervjuer för att fånga kunskapen om standardsystemsutveckling.*

*Resultatet av studien är en utvecklingsmodell som beskriver hur utvecklingen av standardsystem går till och vilka faser den innehar samt hur de avviker från konventionella livscykelmodeller. Det beskrivs även mer ingående vilka faser som ingår i den identifierade utvecklingsmodellen, Förstudie, Planering, Utveckling, Test, Implementering och support. Hur de fungerar och vilken betydelse de har för standardsystemet.*

### Nyckelord

Systemutveckling, Standardsystem, livscyklar, Utveckling.

## Förord

Denna uppsats har skrivits vid en tidpunkt när IT-branschen har haft en tung period men nu kan blicka mot framtiden och återigen utvecklas. Många företag, speciellt mindre företag, har haft det extra svårt och använt sin överlevnadsförmåga till det yttersta för att klara sig undan denna kris. I just denna vändpunkt har vi haft turen att få besöka och studera utvecklingsprocessen hos två IT-företag.

De företag som vi har besökt och som vi vill framhålla i detta förord är dels Microcraft i Borås och dels Pipechain i Lund. De har varit den resurs som har fört denna studie till en högre nivå beroende på den information och kunskap de delgett oss. Ett speciellt tack till produktcheferna på respektive företag, Robert Lindström på Microcraft och Johan Lindell på Pipechain, för den både övergripande och detaljerade bild de givit oss av dessa båda företag.

Sedan vill vi även passa på och tacka vår handledare Jonas Hedman på Institutionen för Informatik vid Lunds Universitet som givit konstruktiva råd och synpunkter under skrivandets gång.

# Innehållsförteckning

<b>1</b>	<b>INLEDNING.....</b>	<b>1</b>
1.1	BAKGRUND.....	1
1.2	PROBLEMFÖRMULERING.....	2
1.3	SYFTE.....	2
1.4	AVGRÄNSNINGAR.....	3
<b>2</b>	<b>METOD.....</b>	<b>4</b>
2.1	VETENSKAPLIGT SYNSÄTT OCH PERSPEKTIV.....	4
2.2	FALLSTUDIE .....	5
2.3	INSAMLING AV DATA .....	5
2.3.1	<i>Intervjuer .....</i>	<i>6</i>
2.3.2	<i>Intervjupersoner.....</i>	<i>6</i>
2.4	ANALYS.....	7
2.4.1	<i>Exempel på hur kodningen gått till .....</i>	<i>8</i>
2.4.2	<i>Validitet.....</i>	<i>10</i>
2.4.3	<i>Reliabilitet.....</i>	<i>11</i>
2.5	KÄLLKRITIK .....	11
2.5.1	<i>Primärdata.....</i>	<i>11</i>
2.5.2	<i>Litteratur.....</i>	<i>12</i>
2.6	ETIK .....	12
<b>3</b>	<b>LITTERATUR .....</b>	<b>13</b>
3.1	LIVSCYKLER.....	13
3.1.1	<i>Vattenfall.....</i>	<i>13</i>
3.1.2	<i>Spiralmodellen .....</i>	<i>14</i>
3.1.3	<i>Livscyklar och standardsystem.....</i>	<i>16</i>
3.2	INFORMATION SYSTEM DEVELOPMENT .....	16
3.2.1	<i>Metoder i praktiken.....</i>	<i>18</i>

3.3	STANDARDSYSTEM.....	19
3.4	SKILLNADER MELLAN ISD OCH STANDARDSYSTEMSUTVECKLING .....	20
3.5	FRÅN LITTERATUR TILL RESULTAT .....	22
<b>4</b>	<b>RESULTAT .....</b>	<b>24</b>
4.1	PRODUKTLIVSCYKELN.....	24
4.1.1	<i>Produktens spridning</i> .....	25
4.2	UTVECKLINGSPROCESSEN .....	27
4.2.1	<i>Förstudie</i> .....	28
4.2.1.1	Användarförening .....	30
4.2.1.2	Kundbesök .....	30
4.2.1.3	Support.....	30
4.2.1.4	Önskemål .....	30
4.2.1.5	Säljmöten och systempresentationer .....	30
4.2.1.6	Tekniska framsteg.....	31
4.2.1.7	Krav och standards.....	31
4.2.1.8	Lagar och förordningar .....	31
4.2.2	<i>Planering</i> .....	32
4.2.2.1	Kravspecifikation.....	33
4.2.2.2	Risikanalys.....	33
4.2.2.3	Testspecifikation .....	34
4.2.2.4	Utvecklingsspecifikation.....	34
4.2.3	<i>Utveckling</i> .....	35
4.2.3.1	Modellering.....	36
4.2.3.2	Dokumentation.....	36
4.2.3.3	Systemering .....	36
4.2.4	<i>Testning</i> .....	37
4.2.4.1	Testning mot testspecifikation .....	38
4.2.4.2	Testning hos kund.....	38

4.2.4.3	Integrationstest.....	39
4.2.4.4	Funktionella tester.....	39
4.2.4.5	Test i egen verksamhet.....	39
4.2.4.6	Test mot kravspecifikation.....	39
4.2.5	<i>Implementering</i> .....	40
4.2.5.1	Implementation av extern part.....	41
4.2.5.2	Implementation av intern part.....	41
4.2.6	<i>Support</i> .....	42
4.2.6.1	Uppdateringar.....	43
4.2.6.2	Informationsutskick.....	43
4.2.6.3	Telefonsupport.....	43
4.2.6.4	Ta emot krav, önskemål och felrapporter.....	44
4.3	UTVECKLINGSLIVSCYKELN.....	44
<b>5</b>	<b>DISKUSSION OCH SLUTSATS.....</b>	<b>46</b>
5.1	REFLEKTIONER.....	48
5.2	FÖRSLAG TILL FRAMTIDA FORSKNING.....	48
	<b>BILAGA 1: INTERVJUGUIDE.....</b>	<b>50</b>
	<b>BILAGA 2: CAUSAL NETWORK.....</b>	<b>53</b>
<b>6</b>	<b>KÄLLFÖRTECKNING.....</b>	<b>54</b>

# 1 Inledning

*I detta kapitel ges en introduktion till uppsatsämnet samt en bakgrundsbeskrivning och en problemformulering. Detta leder fram till syftet med uppsatsen och dess avgränsningar.*

Denna uppsats tar i anspråk processen för hur standardsystem utvecklas. Det kommer att beskrivas hur denna utveckling skiljer sig från utvecklingen av konventionella system. Den initiala skillnaden är att standardsystem utvecklas för en marknad, mer än en kund, medan den konventionella systemutvecklingsmodellen gäller utveckling av system som skall passa en enda kund. Standardsystem kan beskrivas som en färdig programvara som ett företag kan använda i sin egen verksamhet efter att ha gjort mindre anpassningar på systemet. Vidare kan standardsystem beskrivas som ett system som har använts av någon tidigare och som har en redan fungerande infrastruktur (Nilsson, 1991). Konventionella system däremot är system som använder informationsteknologi för att fånga, skicka, lagra, manipulera eller visa information. Konventionella system återfinns i enskilda organisationer och berör personer som är relaterade till organisationen. Utvecklingen av dessa system sker specifikt för den organisation som efterfrågar det konventionella systemet och människorna i denna organisation (Alter, 1999).

Med tanke på att det inte finns så mycket fakta kring hur standardsystem utvecklas i jämförelse med den utförliga, modellstrukturerade utvecklingen som finns för konventionella system så görs det här ett försök att belysa själva processen för utveckling av standardsystem. För att sedan jämföra den med utvecklingen av konventionella system och på så sätt konkretisera vilka utvecklingssteg som sammanfaller respektive avviker har varit ett av målen med denna studie.

För att generera en utgångspunkt för denna studie har valts att genomföra empiriska undersökningar på två företag. Detta för att skapa en bild av hur standardsystem utvecklas och för att ha denna som jämförelsebakgrund.

Den teori som framhålls i denna uppsats gäller huvudsakligen konventionell systemutveckling, som sedan jämförs med empiriskt insamlat material från de valda fallföretagen.

## 1.1 Bakgrund

Företag lever under ständig press att utvecklas och effektivisera sina interna processer, det vill säga hur verksamheten bedrivs, för att vara konkurrenskraftiga och att använda informationsteknologi för att komma till den effektiviseringen blir allt vanligare (Cooper & Zmud, 1990 enligt Child 1987, Drucker, 1988). De standardsystem som implementeras i organisationer är ofta av komplex natur och passar sällan exakt den organisation där de implementeras, vilket gör det extra

## Inledning

### - *Konsten att göra det specifika generellt*

intressant att undersöka hur standardsystem utvecklas, då dess komplexitet antagligen medför ytterligare en dimension i utvecklingsprocessen.

Det var redan i mitten på 1950-talet som organisationer började använda sig av datorbaserade informationssystem. Det var så det hela började, men idén om att det kunde effektivisera verksamheten växte fram först i mitten på 1960-talet (Kwon & Zmud, 1987).

Över tiden började system tillverkas för att stödja olika funktioner inom organisationer, det var allt från lagersystem till lönesystem. Att ett system kan fungera för flera företag är naturligtvis möjligt, men hur skall systemet vara utformat för att det skall passa så många organisationer som möjligt? (Kwon & Zmud, 1987)

Standardsystem har vuxit fram till att vara en väldigt stor del av informationsteknologin. Redan 1998 hade utvecklingen av standardsystem vuxit till att vara den femte största industrin i USA (Sawyer, 2000).

## **1.2 Problemformulering**

Många av de systemutvecklingsmodeller som finns idag har funnits sedan början av 1980-talet och de flesta av dem fokuserar på hur system utvecklas för en specifik kund. Sedan dess har dock mycket hänt inom systemutvecklingen, bland annat har det blivit vanligare att utveckla så kallade standardsystem, system som är tänkta att kunna appliceras på ett flertal olika kunder. Det finns många lyckade sådana system som har gjort organisationer mer konkurrenskraftiga, medan det också har tillfört enorma kostnader för andra. Men trots den uppmärksamhet som ämnet har fått, finns det inte mycket forskning på hur utvecklingen av dessa system faktiskt går till och vad som skiljer denna utveckling från de konventionella metoderna. Utifrån detta kommer uppsatsens tyngdpunkt att ligga på:

*Hur ser utvecklingen av standardsystem ut i idag och hur skiljer den sig från konventionella utvecklingsmetoder?*

## **1.3 Syfte**

Syftet med denna uppsats är att belysa hur utvecklingen av standardsystem går till i praktiken. Att berika forskningen kring utveckling, med kunskap kring och om hur utvecklingen av standardsystem går till.

Syftet med uppsatsen är även att förstå hur utvecklare arbetar för att göra de processer som är specifika i olika fall och miljöer generella. Så generella att de kan innefattas i ett system som kan användas av många företag och i många miljöer. Detta delvis genom att konkretisera standardsystemsutvecklingen jämfört med konventionell systemutveckling. Uppsatsen är i första hand avsedd för studenter och lärare i informatik och relaterade ämnesområden, samt andra med



intresse för utvecklingsprocessen gällande standardsystem. Förhoppningsvis kommer denna uppsats att motivera och uppmuntra till vidare undersökningar och studier inom området.

### **1.4 Avgränsningar**

En av avgränsning gjorda för denna uppsats är att endast undersöka hur utvecklingen av standardsystem går till på två utvalda fallföretag. Detta på grund av att en undersökning som omfattar fler företag skulle uppta allt för stora resurser, och därför inte få det djup som här eftersträvas. Att endast fokusera på två fall ger istället möjligheter att studera och analysera dessa fall djupare och på så vis få en bättre förståelse för ämnesområdet.

Vi har även avgränsat denna uppsatts genom att utgå endast ifrån utvecklingen av standardsystem och då helt utelämna andra aspekter som skulle kunna vara intressanta, som till exempel användandet av standardsystem eller val av standardsystem.

Avgränsningar har även gjorts för att endast röra de delar som faller under vad som händer på utvecklingsföretagen. Faktorer som rör andra organisationer som blir påverkade av standardsystemsutvecklingen har inte valts att tas upp i denna uppsatts. Faktorer som är rent organisatoriska på fallföretagen har heller inte valts att diskuteras inom ramarna för denna uppsats då det anses ligga utanför studiens syfte.

## 2 Metod

*I detta kapitel kommer det att diskuteras hur studien har gått tillväga när undersökningen genomfördes, även metodval kommer att diskuteras i detta kapitel. Det kommer endast diskuteras det tillvägagångssätt som valts och ges ingen djupare inblick i alternativa arbetssätt.*

### 2.1 Vetenskapligt synsätt och perspektiv

Som grundsyn för denna uppsats har valts ett hermeneutiskt förhållningssätt. Hermeneutik som betyder tolkningslära, och är positivismens raka motsats, fokuserar på att inhämta kvalitativ förståelse och tolkning (Wallén, 1993). I det Hermeneutiska förhållningssättet försöker forskaren vara subjektiv och engagerad. Det är även så att forskarens mänskliga existens med inbäddade erfarenheter, värderingar och normer påverkar undersökningens resultat (Patel & Davidson, 1991).

Som beskrivits tidigare vill denna uppsats fånga hur utvecklingsprocessen av ett standardsystem går till. Varpå det var lämpligt att genomföra en kvalitativ studie som fångar intervjuobjektens erfarenheter gällande den egna utvecklingsprocessen. För att kunna studera komplexa problem som är svåra att mäta med något mätinstrument behövs alltså kvalitativa undersökningsformer (Wallén, 1993). Bryman (1997) framhäver också att det kan vara viktigt att förstå människors tolkningar av fenomen eller händelser, för att skapa kunskap om innebörden av det mänskliga beteendet i olika situationer, vilket i detta fall är intressant då det är utvecklarens syn på händelseförloppet i utvecklingsprocessen som kommer studeras.

Det har vid efterforskningar gällande uppsatsens ämnesområde inte funnits något vetenskapligt material som behandlar utvecklingsprocessen för standardsystem specifikt. Mycket litteratur har dock funnits i närliggande ämnen, såsom systemutveckling, standardsystem, livscyklar med mera. Bristen på information om det specifika ämnet gör därför att det kan anses som ett utforskat område rent vetenskapligt, varpå det faller sig naturligt att anta en explorativ ansats. Detta innebär att undersökningen blir utforskande och att det främsta syftet blir att inhämta specifik kunskap inom det aktuella området för att allsidigt belysa problemställningen (Patel & Davidson, 1991). Studiens utgångspunkt har varit oprecisa och vaga föreställningar, det vill säga att det inte har funnits några hypoteser eller förutfattade meningar om uppsatsämnet i begynnelsefasen.

Denna studie kan främst beskrivas som induktiv, detta på grund av det empiriska materialet som samlas in, genom en fallstudie, för att sedan sammanställas och analyseras för att utmytna i en hypotes om hur utvecklingen av standardsystem går till (Wallén, 1993).

## **2.2 Fallstudie**

I undersökningen har det valts att arbeta med metoden fallstudie som ses som den vanligaste formen av kvalitativ studie (Bryman, 2003), och som innebär en fokusering på ett fåtal objekt i en mängd avseenden (Eriksson & Wiedersheim-Paul, 1998). Detta har valts eftersom studien syftar till att samla in information gällande hur utvecklingen av standardsystem sker på två företag. Då det inom detta ämne finns en brist av kunskaper i form av böcker, artiklar och annan vetenskaplig litteratur, så kommer det underlätta att arbeta med tillvägagångssättet av en fallstudie för att komma ut i verkligheten och samla in den kunskap som finns hos de som arbetar med utveckling av standardsystem. Det har valts att genomföra en fallstudie som fokuserar på studier i två olika organisationer, detta i ett försök att ge bättre generaliserbarhet för de resultat som studien utmynnar i (Miles & Huberman, 1994).

En fallstudie kan delas upp i två olika kategorier, embedded case study och logistic case study (Yin, 2003). I undersökningen gjordes valet att använda embedded case study, eftersom det finns ett tydligt formulerat problem och ett tydligt syfte med uppsatsen. De två företagen som undersöks utvecklar båda två standardsystem vilket gör att de faller under samma kontext och möjliggör att använda flera analysenheter inom samma fall.

Fallstudien har valts att genomföras på företag som arbetar med utveckling av standardsystem, och som har mångårig erfarenhet gällande utvecklandet av denna form av system. Att de arbetat med utveckling av standardsystem under en längre tid ger fördelen av att det på fallföretagen finns en stor kunskap om hur utvecklingsprocessen förlöper.

De företagen som valdes att förlägga fallstudien till var Microcraft som har sitt kontor i Borås, och Pipechain, baserat i Lund. Microcraft har funnits sedan början 80-talet och är en organisation med ungefär 40 personer anställda som utvecklar ett affärssystem som heter GARP. Produkten är riktad mot små och medelstora företag och har ungefär 1900 kunder. Pipechain har en något mindre organisation på ca 20 anställda och har existerat som eget bolag sedan 1999, men har tidigare legat som en egen organisatorisk enhet under MA-system sedan 1997. Deras standardsystem skiljer sig från GARP genom att det är ett mindre mer specialiserat logistik system och som är riktad mot i första hand större företag. Pipechain har 40 kunder, men systemets huvudkunder består i dagsläget av Ericsson, Volvo och SCA.

## **2.3 Insamling av data**

Valet av att arbeta med intervjuer har gjorts eftersom det är viktigt för studien att få en djup förståelse för hur utvecklingsarbetet fungerar, då detta är ett av delmålen med studien. Användning av muntlig intervju istället för skriftlig, i form av en enkät, kommer av dess fördelar att kunna upprepa och förtydliga frågor, ställa följdfrågor och förvissa sig om att ha fått uttömmande svar (Andersson, 2001). Det finns även en medvetenhet om att denna studies val att arbeta endast

med två fall medför svårigheter att generalisera resultatet. Detta är en av anledningarna till att det valts att arbeta utifrån intervjuer istället för till exempel observationer (se kap2.3). Intervjuer har fördelen av att kunna ge en bredare bas av respondenter från olika sociala och geografiska miljöer, till skillnad från observationerna som endast fokuserar på en situation (Bryman, 1997).

### **2.3.1 Intervjuer**

Vi har valt att genomföra sju intervjuer, uppdelat på två stycken fallföretag, vilket gav ett sammanlagt intervjumaterial på ungefär tretton timmar, som anses ha gett en god grund för slutresultatet av studien. Intervjumetoden som användes var semistrukturerade intervjuer. Dessa genomfördes med personal på fallföretagen i deras normala arbetsmiljö. Detta för att få en inblick i deras arbete och på så sätt även få en bild av hur utvecklingsarbetet fungerar. Målet med detta var att möjliggöra intervjuer under avslappnade former, vilket förhoppningsvis genererade bättre svar på de frågor som ställs och därmed en bättre grund för ett bra slutresultat.

Att arbeta med semistrukturerade intervjuer har valts eftersom det anses kunna ge en mer avslappnad intervjuform som till stor del kan liknas vid ett samtal. Den ger även möjligheter för samtalet att gå sin egen väg och komma fram till saker som inte hade framkommit vid en strukturerad intervju. Valet att arbeta med en semistrukturerad intervjuer kan gynna uppsatsen då en avslappnad intervjuform på ett bättre sätt kan få intervjupersonerna att prata obehindrat. En ostrukturerad intervju ställer däremot större krav på forskarnas förmåga och erfarenhet som intervjuare (Andersson, 2001), och då det saknades praktiska erfarenheter av intervjusituationer valdes det i studien att inte arbeta på detta vis.

Nackdelen med den semistrukturerade intervjun är risken att tappa den röda tråden i samtalet så att den leder in på för uppsatsen irrelevanta områden. Detta försökte motverkas till viss del genom att ha en detaljerad intervjuguide (bilaga 1), samt genom att vara så väl förbered och påläst inom de ämnesområden som ligger nära det undersökta. Intervjuguiden är baserad på vad som framkommit i förstudien (kapitel 3) och användes som ett hjälpmedel för att kunna fokusera intervjun på rätt saker, samtidigt som den gav möjligheter att gå iväg på tillfälliga relevanta sidospår för att sedan återgå till intervjuguiden.

### **2.3.2 Intervjupersoner**

De intervjupersoner som valdes för studien har alla viktiga roller och stor kunskap inom den egna utvecklingen. Det ansågs att vikten av att välja rätt respondenter var stor då det inte fanns möjlighet att intervju alla personer på företagen involverade i utvecklingsprocessen. Anledningen till detta var resurser för den här studien och brist på resurser som företagen var beredda eller hade möjlighet att avsätta.

Valet av respondenter gjordes således i samråd med produktcheferna på respektive företag. Detta genom att en lista på önskvärda respondenter lämnades in till produktchefen för att sedan diskuteras runt. I stort sett blev önskemålen

tillgodosedda utom i de fall där denna individ inte kunde frigöra tid för detta. Att produktchefen var inblandad i valet av respondenter var en nödvändighet för att intervjuerna skulle kunna genomföras på önskvärt sätt, men kan ha medfört att en förbättrad bild av fallföretagen kan ha framkommit, genom att endast de mest positiva medarbetarna valts. Det kunde då garanteras att personer som tillsammans kunde ge en bred och djup bild av utvecklingsprocessen blev respondenter för studien.

Det valdes att förlägga intervjuerna under två dagar på respektive företag. Detta i ett försök att få intervjuerna att omfatta hela utvecklingsarbetet, ge möjlighet till reflektion efter hälften av intervjuerna och då få uttömmande svar på de frågor som ställdes. De respondenter som studien haft har omfattat ett flertal olika områden som är i kontakt med utvecklingsprocessen, såsom produktchef, projektledare, programmerare och säljare.

Vid intervjusituationerna har det i de flesta fall varit två intervjuare och en respondent, i två fall endast en intervjuare. Det kan anses att det i de fall där det varit två intervjuare skulle finnas en maktobalans. Det faktum att det har varit två forskare på ena sidan och en professionell systemutvecklare på andra sidan har gjort att maktbalansen vägts upp en aning. Att intervjuerna sedan har varit förlagda till utvecklarens normala miljö anses ha vägt upp den maktobalans som annars kunnat vara svårhanterlig.

## **2.4 Analys**

Vid undersökningen har analysen funnits i åtanke redan innan intervjuerna genomfördes. Detta för att undvika en situation med ett stort antal transkriberade intervjuer utan någon strategi för hur dessa skall analyseras, något som för många kan skapa stora problem (Kvale, 1997). Genom att ha analysen i åtanke redan vid genomförandet av intervjuerna kunde analysprocessen påbörjas redan i själva intervjusituationen (ibid). Något som var viktigt för studien då den hade ett omfångsrikt intervju material och ett pressat tidsschema. Att arbeta på detta vis, med en delvis integrerad analys och datainsamling gör också att det på ett snabbt och enkelt sätt går att fylla de eventuella kunskapsluckorna som kan ha uppstått under datainsamlingen. Att arbeta med att först göra insamling och sedan analys minskar möjligheterna att kunna komma tillbaka på ett naturligt sätt till undersökningsfallet och genomföra ytterligare studier efter det att den egentliga undersökningen är avslutad (Miles & Huberman, 1994), vilket är fallet med de båda företagen som undersökts då det i deras fall handlar om resurser i form av tid som de inte kunde sätta till förfogande i allt för stor utsträckning.

Analysarbetet har delats upp i tre faser, data reduktion, data uppvisning och slutsatsdragning/verifikation (Miles & Huberman, 1994). Den data som samlats in genom intervjuer har transkriberats med relativt hög detaljnivå för att till en början inte tappa alltför mycket information och för att underlätta memorering, tolkning och förståelse av innebörden av de uttalanden som gjorts. Detta gjordes för att skapa en grov grund till en mer omfattande reduktion och förenkling av den data som samlats in.

Sammanställningen och uppvisningen av data underlättades genom kodning av de transkriberade intervjuerna för att på så vis få en mer organiserad bild av den data som samlats in. För kodningen av den insamlade datan användes ”pattern coding” (Miles & Huberman, 1994) som möjliggjorde att datan kunde delas upp i mindre delar och gemensamma teman och på så vis ge den en bättre struktur vilket underlättade arbetet med att se samband och dra slutsatser ur den insamlade datan.

För att ytterligare strukturera datan och för att göra det enklare att dra slutsatser ifrån densamma har arbetats med att strukturera data visuellt. Denna strukturering åstadkoms genom användning av ”displays” som enligt Miles & Huberman (1994) hjälper forskare att skapa förståelse för samband mellan händelser, flöden och plats. Vilket lämpar sig för fall där det främsta arbetet kommer att bestå av att försöka förstå samband mellan olika händelser, och som i denna studie kommer väl till hands för att förstå utvecklingsprocessen. Typen av display som använts är ”networks” eftersom denna display bygger på att det skall gå att se en löpande ordning av händelser och skeenden, till exempel för en process, men den ger även möjligheten att särskilja de olika händelserna i nätverket med ytterligare förklarande text (ibid). Ännu mer specifikt är det ett kausalt nätverk som har använts (bilaga 2), då denna typ av nätverk gör det möjligt att dela in flödet av händelser i olika faser på ett visuellt strukturerat sätt (ibid).

#### **2.4.1 Exempel på hur kodningen gått till**

Här nedan kommer det att ges ett exempel på hur studien gått tillväga från transkript till analyserbart material. Exemplet visar hur kodningen gått till av de 115 sidor transkriberat material som genererats från intervjuerna, samt hur det sedan presenterats i det kausala nätverket. Men för att kunna utföra kodningen behövdes ett underlag för denna som konstruerades efter de olika faserna som presenteras i kapitel 3.5. Utöver de identifierade faserna lades till kodning för den specifika fallstudien (Case) och övriga delar (Övrigt) som skulle kunna berika studien. Detta underlag fick följande utseende:

Case	C
-----	
Förstudie	F
Planering	P
Utveckling	U
Testning	T
Implementering	I
Support	S
-----	
Övrigt	Ö

Här följer ett utdrag av transkriptet och det visas även hur kodningen resulterade på detta avsnitt:

Metod  
- *Konsten att göra det specifika generellt*

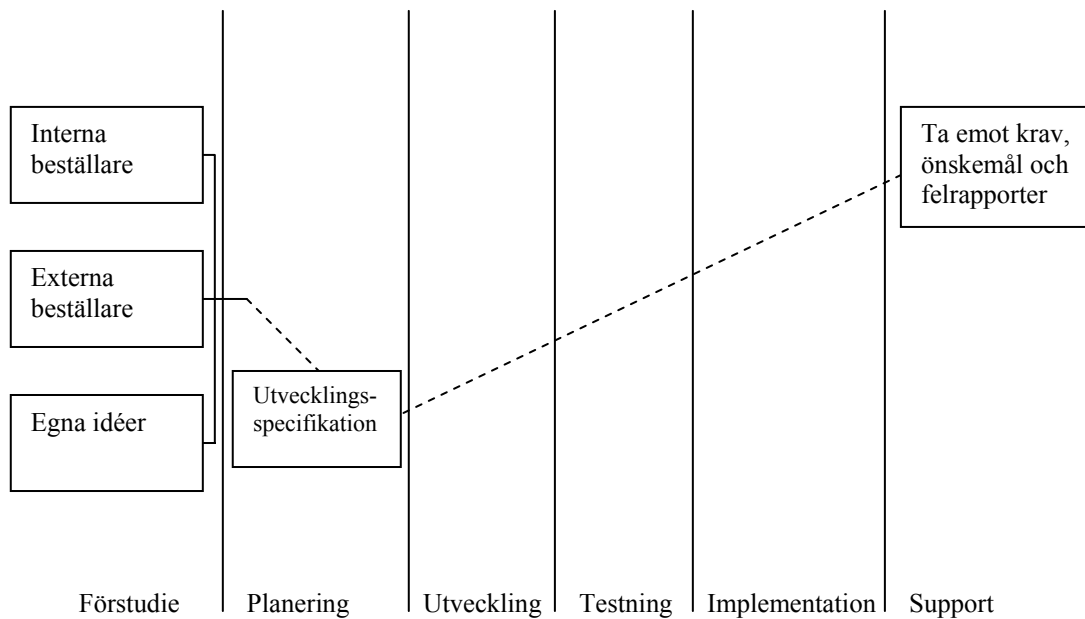
**Intervjuare:** *I inledningen tas det fram en kravspecifikation baserad på dels användarföreningen och dels det befintliga systemet.* (F)

**Respondent:** *Det finns ett antal olika beställare då, interna beställare som får signaler från marknaden till exempel säljare som hittar nya behov som vi måste uppfylla. Och så har vi kunderna då. Sen har vi ju också andra interna beställare då, i form av de som sitter och tar emot signaler från kunderna det vill säga supportpersonal. Det kanske inte är signaler från en enskild kund utan det sammantagna intrycket som man får. Alltså två typer av interna beställare och så har vi ju kunderna som är beställare i det dagliga då. Eh... kan man säga faller in under första punkten, fast det kanske inte är säljare då utan supportpersonal och utvecklare som är ute och ser att det finns nya möjligheter. Och det i sig skapar nya krav. Framförallt för att man ser att det finns nya möjligheter. Kunderna har inte i sig lösningskraven utan när de ser att man kan komma åt saker externt för att skapa nya rutiner för detta tycker vi att det är kanon, men det kanske krävs att det finns en förståelse och öppenhet hos utvecklare och tekniker som kan bedöma tekniker som är möjliga och använda och till exempel se att nu kan man öppna gränssnittet då... det finns ett antal olika kanaler, beställare i första fasen.* F  
S  
F  
P

Kodningen skedde på så vis att hela transkriptet kodades på två separata håll och sedan synkroniserades de två kodningarna för att kontrollera så att samma åsikter hade framkommit gällande vad olika uttalanden hade för betydelse. Det visade sig ha stor samstämmighet vilket gav tyngd åt det kodade materialets kvalitet.

Ifrån det kodade transkriptet registrerades de delar som belagts med koder och lades in i det kausala nätverket som separata delar under den rubrik som överensstämde med kodningen enligt figur 2.1 nedan.

Metod  
- *Konsten att göra det specifika generellt*



**Figur 2.1. Från kodning till Kausalt Nätverk**

För att sedan generera överskådlighet har vissa delar slagits ihop, till exempel har Externa och Interna beställare slagits samman till Kundförfrågan vilket visas i bilaga 2 tillsammans med resten av resultatet från kodningen. I figur 2.1 finns det streckade linjer som symboliserar att det finns objekt på vägen som inte framkommit av det exempel som visats.

## 2.4.2 Validitet

Med validitet avses huruvida studien mäter det som avsågs att mätas. Vanligtvis skiljs på yttre validitet och inre validitet. Yttre validitet syftar till forskningens validitet utanför experimentsituationen och hur pass tillämpningsbara resultaten är på andra situationer. I detta fall, hur pass väl de utvecklingsmetoderna som studerats på fallföretagen stämmer överens med den generella utvecklingsprocessen för standardsystem som beskrivs i olika forskningsteorier. Det vill säga inte generaliserbart mot en population utan mot en tillgänglig samling teorier. Inre validitet innebär hur pass väl empiri och teori stämmer överens och hur upplägget på undersökningen ser ut (Yin, 2003).

Validitet i de instrument som använts i forskningen bör också granskas kritiskt, vilket i kvalitativ forskning innebär granskning av forskarens färdigheter och förfrågningar om forskaren i fråga är ett bra forskningsinstrument. Några exempel på vad som gör en bra forskare i detta fall kan vara att denne är någorlunda bekant med fenomenet, har en multidisciplinär ansats och bra undersökningsfärdigheter (Miles & Huberman, 1994). För att stärka forskarens validitet i intervjuerna har forskarna i denna undersökning varit pålästa och förberedda samt utfört testintervjuer. Testintervjuerna syftar till att stärka förmågan att ställa relevanta och genomtänkta frågor samt berikande följdfrågor.



Respondentens validitet säkerställs genom att efter intervjuernas utförande kontrollera resultatet med dem som har undersökts. Nackdelen med detta tillvägagångssätt kan vara att då känslig information framkommit kan de undersökta personerna vilja censurera just denna information som tidigare berikade studien. Det har i denna studie ändå valts att användas, detta då risken för att stöta på känslig information som respondenterna skulle vilja censurera bedömts som relativt liten. Nyttan av att på ett bra sätt kunna bli försäkrad om att den insamlade datan är korrekt bedöms vara tillräckligt stor för att motivera detta beslut.

### **2.4.3 Reliabilitet**

Med reliabilitet menas tillförlitlighet till forskningsstudien. Med detta brukar menas huruvida undersökningen har utförts på rätt sätt, om det ställts rätt frågor, använt sig av rätt metoder och undersökt rätt personer så att samma resultat skulle ha genererats om undersökningen skulle ha upprepats. När det pratas om reliabilitet handlar detta om konsistensen som utgörs av två olika dimensioner. Inre konsistens som innebär de mått som kan ta formen av en skala eller ett index. Den andra dimensionen rör konsistens för ett mått över tid. Om detta mått skall bestämmas måste mätningen göras mer än en gång. Det tenderar att vara så att forskare är mer intresserade av reliabilitet än av validitet då validitetsprövningen kan ta väldigt lång tid (Bryman, 1997).

Att gå tillbaka till respondenterna och kontrollera den fakta som framkommit kommer även att användas för att stärka reliabiliteten i forskningen då respondenterna kommer att få chansen att kritisera de metoder som använts och den information som undersökningen resulterat i.

## **2.5 Källkritik**

Att forskaren påverkar studien och att denne inte alltid kan förhålla sig objektiv till den egna studien är något som Gilovich (1991) tar upp. Detta förstärks med Denscombe (1998) som anser att forskarens "jag" spelar en viktig roll i produktionen och tolkningen av kvalitativa data. Syftet med att använda källkritik är att införa en medvetenhet om källans relevans för studien.

Det är inte lätt att avgöra på vilket sätt undersökningen påverkas, men genom att ständigt reflektera över och ifrågasätta resultatet kan en alltför stark prägling av slutresultatet undvikas.

### **2.5.1 Primärdata**

Primärdatan till denna undersökning består av intervjuer med personer på ett flertal olika positioner i de organisationer som undersökts. De har alla viktiga positioner i utvecklingsprocessen av standardsystem och dessa personer besitter både erfarenhet och djupare kunskaper om standardsystem vilket gjort dem till, för studien, lämpliga intervjuobjekt. Det har vid analysen av intervjuerna fokuserats på hur de olika respondenternas syn på saker interagerar med varandra.

## 2.5.2 Litteratur

Den litteratur som används i denna uppsats är huvudsakligen hämtad från vetenskapliga artiklar. Böcker har också använts, dock i mindre utsträckning då informationen i dessa ofta har gått att ersätta med mer aktuell information funna i artiklarna. Att de litterära källorna är aktuella är naturligtvis viktigt när detta används för att analysera det empiriska materialet. Målet med den litteratur som använts har därför varit att ha den mest aktuella eller senaste forskningen tillgänglig vid analysen. Dock finns det fall där lite äldre litteratur har använts, dock endast när nyare information inom ämnet inte har gått att finna. Att det funnits en brist på forskning inom vissa ämnesområden har även lett till att viss information inte har kunnat få stöd fullt ut från ett flertal kompletterande källor, vilket hade varit att föredra. Men detta endast i fall där inga kompletterande källor funnits att tillgå.

## 2.6 Etik

För att medvetandegöra de etiska synvinklar som finns kring undersökningen har Miles & Huberman (1994) dragit upp ett antal riktlinjer, de av dessa som ansetts relevanta diskuteras här nedan.

Vid undersökningen har syftet varit ganska så klart och har tydligt förmedlats till dem som undersöks då detta tillvägagångssätt snarare kan stärka än försvaga möjligheterna att lyckas med undersökningen. På detta vis kan en del av misstänksamheten som kan finnas mot forskare undermineras och då ge respondenterna trygghet genom att ge dem förståelse om att det endast är processen som studeras, inte de intervjuade. Genom att göra de som undersöks medvetna om varför de studeras undviks också att de skulle kunna känna sig kränkta av att inte ha fått veta hela sanningen om undersökningen.

I undersökningsarbetet och presentationen av materialet har det tagits hänsyn till personlig integritet och därav har en avpersonifiering av det material som använts i uppsatsen gjorts. Detta för att kommentarer och åsikter inte ska kunna härledas tillbaka till en specifik person då detta eventuellt kan leda till att någon enskild person pekas ut eller blir drabbad.

Det har även tagits hänsyn till att uppsatsen kommer att bli offentlig handling och att den på så vis skulle kunna användas av olika personer i olika kontexter. På grund av detta har det varit viktigt att tänka efter vilka som kan komma att använda publikationen och till vad, och även vilka som kan komma till skada av vad som framkommit av undersökningen. Det är bland annat viktigt för studien att skydda fallstudieföretagen så att de inte kommer till skada av vad som publiceras om dem, och därför tänka efter innan publicering av information som kan vara känsliga för dem sker.

Det har även tagits hänsyn till att forskare har en etisk skyldighet i att presentera korrekt information och därför inte kan manipulera data för att resultaten som genereras skall bli de som avsetts att generera, eller att undanhålla känslig information på något vis som är viktigt för studien.

### 3 Litteratur

*I detta kapitel kommer det att framhåvas olika teorier från den litteratur som uppsatsen bygger på. Inledningsvis kommer det att presenteras hur olika livscyklar fungerar och vilken betydelse de har. Det presenteras livscykeltänkande och det ges även exempel på hur dessa livscyklar är utformade och hur pass väl de passar in på standardsystemsutveckling, detta för att bygga en grund för processen bakom standardsystemsutveckling senare i uppsatsen. Därefter kommer konceptet Information System Development (ISD) att presenteras och beskrivas. Slutligen kommer det att ges en bred inblick i vad standardsystem är och vilka typer av standardsystem som finns att tillgå. En kortare bild av vad som kan vara specifikt för standardsystemsutveckling kommer även att ges.*

#### 3.1 Livscyklar

Processen bakom utformandet av informationssystem beskrivs ofta i form av en livscykel, och det har skapats ett flertal sådana modeller för att beskriva denna process på olika sätt.

Fokus på utvecklingsprocesser startade redan i början av 1970-talet då det insågs att för att uppnå den kvalitet som eftersträvades var utvecklare tvungna att förstå processen och arbeta efter aktiviteter som var ordnade i en viss följd. Att arbeta efter en fördefinierad process skulle även kunna göra processen mer förutsägbar och ekonomisk. På 1980-talet började forskningen på hur dessa processer ser ut ta fart på allvar och har sedan dess betraktats som en egen vetenskap. Med processen syftas till hur den färdiga produkten skapas. (Cugola & Ghezzi, 1998)

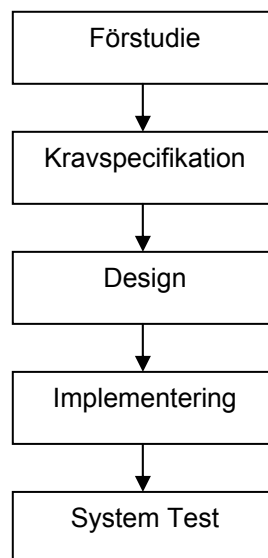
Konventionella livscykelmodeller för systemutveckling (ISD, vidare beskrivet i kapitel 3.2) kan enligt Carmel (1993) kategoriseras enligt följande; vattenfall, inkrementa och spiraler, varav det har valts att vidareutveckla två av dessa, nämligen vattenfall och spiralmodeller eftersom dessa kan sägas vara övergripande modeller som innefattar stora delar av olika utvecklingsprocesser.

##### 3.1.1 Vattenfall

Vattenfallsmodellen (figur 3.1) är antagligen den mest välkända av alla livscykelmodeller och skapades redan på 1970-talet, men den är också antagligen den mest idealiserade av dem. Modellen är baserad mycket på klassiskt ingenjörstänkande och påminner mycket om processer inom tillverkningsindustri där allt sker steg för steg i en viss bestämd följd och där varje steg ska vara färdigt innan nästa steg kan påbörjas. Tanken bakom denna livscykelmodell är alltså att processen sker linjärt, helt utan cykler. Återvinning ses i vattenfallsmodellen som något som inte bör användas, beroende på att varje steg måste tillföra sin del och därför inte kan hoppas över. Efterföljs inte detta kan det uppmuntra till ett slarvigt beteende som gör att medarbetarna inte tänker efter ordentligt innan varje uppgift utförs. Att det i livscykeln arbetas linjärt stöder också tänkandet om att det ska vara lätt att förutsäga och kontrollera processen. Om inte processen är linjär så skapas möjligheter att gå fram och tillbaka mellan olika steg i processen, vilket i

sin tur gör det svårt att kontrollera processen och veta hur långt projektet egentligen kommit (Cugola & Ghezzi, 1998). Det kan sägas att den på detta vis gör det lätt för den som planerar processen, men svårt för de som ska arbeta med den (Kruchten, 2000).

Många företag använder fortfarande den här modellen, men väldigt få använder den i dess strikta originalform då detta visat sig i det närmaste omöjligt att göra på ett bra sätt i praktiken. Ett problem som ofta är återkommande med vattenfallsmodellen är att det är väldigt sällan som det finns klart definierade problem redan från början vilket gör att arbetet ibland måste gå tillbaka i processen. Att följa vattenfallsmodellen slaviskt och därför inte ta hänsyn till fall där det måste tas ett steg bakåt i processen kan ge större problem senare genom att till exempel kostnaderna för att underhålla och vidareutveckla produkten kan bli mycket större. (Cugola & Ghezzi, 1998) Mycket av de justeringar som behövs göras i systemet kommer alltså genom vattenfallsmodellen som en irriterande och ofta oplanerad effekt på slutet av processen. I värsta fall blir slutsatsen att det gjorts misstag tidigare i processen som kan vara svåra att rätta på ett bra sätt (Kruchten, 2000).



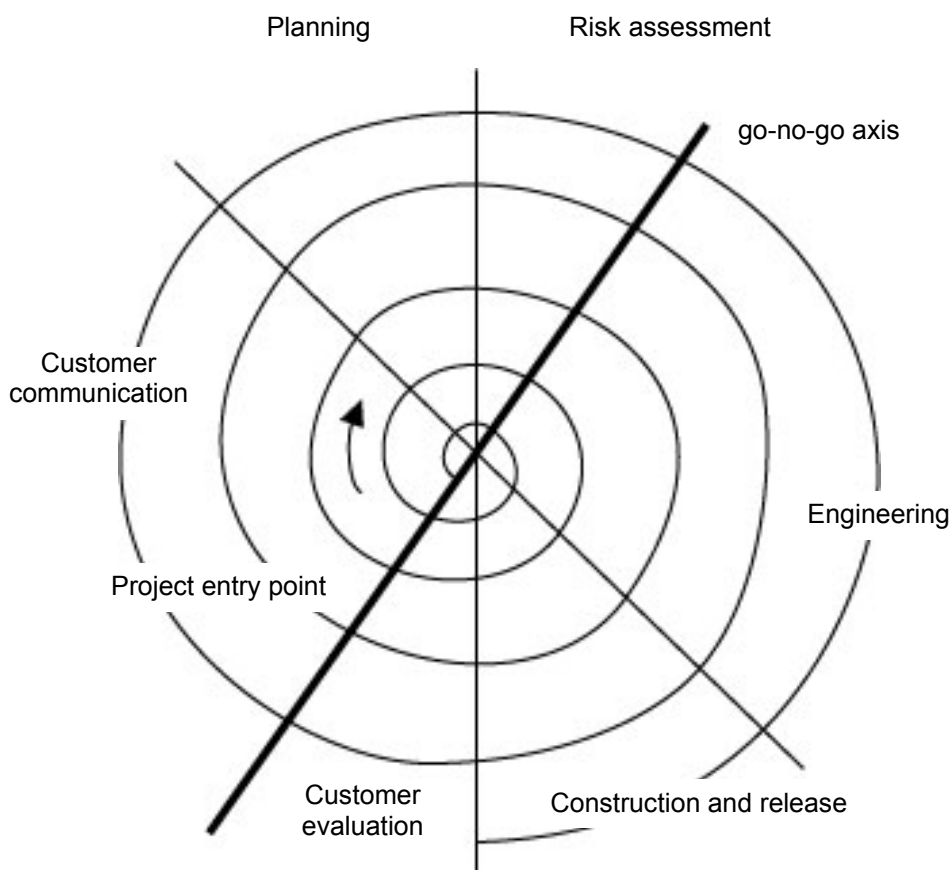
Figur 3.1 Vattenfallsmodell (Cugola & Ghezzi, 1998)

### 3.1.2 Spiralmodellen

En variant att visualisera systemutvecklingsprocessen som skiljer sig från vattenfallsmodellen är spiralmodellen (figur 3.2), som även kallas för en evolutionär modell, den kombinerar det iterativa arbetet som sker i en systemutvecklingsprocess med det systematiska stegvisa tillvägagångssättet som finns i vattenfallsmodellen (figur 3.1). I denna spiralmodell ses utvecklingen av ett system som ett stegvis utvecklande av nya versioner av systemet. I början kan de första versionerna av systemet till exempel röra sig om prototyper, men dessa

byggs senare på under utvecklingen för att bli mer kompletta versioner av systemet. (George, 2000)

Enligt spiralmodellen så går varje varv runt spiralen genom sex olika faser. Den första fasen är "customer communication" som innebär att det ska etableras en god kommunikationskanal mellan utvecklarna och de blivande användarna och/eller kunderna. Den andra delen är "planning" som bygger på att det här sätts upp riktlinjer för projektet och saker som projektets tidsschema och resurser ses över. Nästa fas i processen är "risk assessment", där ses tekniska och organisatoriska risker över. Är riskerna för stora kan processen avbrytas vid "go-no-go axis". Den fjärde fasen är "engineering" som är själva konstruktionen av applikationen (design fasen i vattenfallsmodellen, se figur 3.1). Den femte fasen är "construction and release" som omfattar till exempel tester, installationer och användarsupport. Den sjätte fasen är "customer evaluation" och syftar till att ge feedback på faserna "engineering" och "construction and release". (George, 2000)



**Figur 3.2** Spiralmodellen (Pressman (1976) enligt George (2000))

Spiralmodellen är en mer komplex bild av hur systemutvecklingsprocessen ser ut jämfört med till exempel vattenfallsmodellen, men den ses även som mer realistisk. I mångt och mycket så liknar flera av faserna i spiralmodellen de som finns i vattenfallsmodellen, men har utökats med bland annat en mer realistisk syn på underhåll av systemet, iteration och riskanalys. Spiralmodellen har även varit framgångsrikt använd av flera projekt, men är inte anpassad för att gälla alla projekt då modellen är byggd för att användas i första hand på stora projekt. (George, 2000)

### **3.1.3 Livscyklar och standardsystem**

Dessa olika livscyklar är inte direkt skapade för standardsystemstillämpningar och går därför inte att tillämpa fullt ut i standardsystemsutveckling av olika anledningar. Vattenfallsmodellen gör till exempel ett antagande att kraven kan formuleras exakt och fullständigt. Att dokumentera allting som händer i slutskedet av varje steg kommer också i ett standardsystemsprojekt att resultera i en otroligt stor mängd pappersarbete. Metoden saknar även bra metoder för att generera feedback som kan förbättra arbetet utan tillgång till användare. (Carmel, 1993)

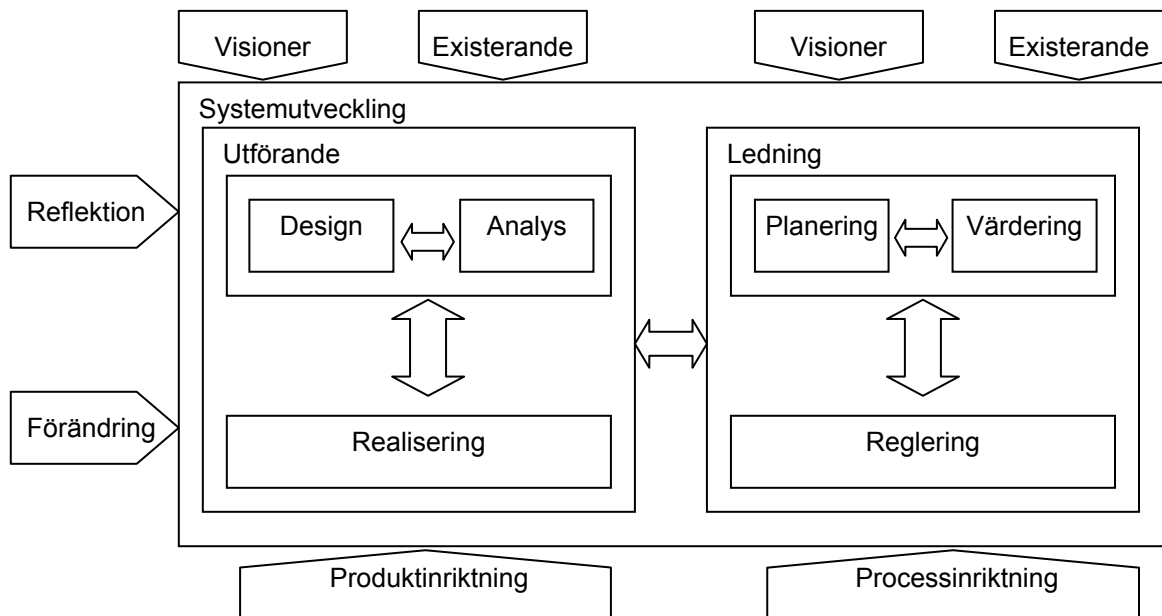
Spiralmodellen är enligt Carmel (1993) den modell som borde vara lättast att anpassa och använda för standardsystemsutveckling eftersom att denna omfattar delvis det stegvisa tänkandet men tar samtidigt hänsyn till riskreduktion. Men spiralmodellen är ändå en modell som är framtagen för att användas för konventionell systemutveckling och så är den anpassad för framförallt större organisationer. Standardsystemsutveckling sker främst i mindre organisationer vilket gör att kompbiliteten av modellen för standardsystemsutveckling blir ganska dålig.

## **3.2 Information system development**

Konceptet *Information System Development* (ISD) beskriver utvecklingen av informationssystem och hur dessa implementeras i organisationer. ISD beskriver även den initiala fasen, hur systemkraven definieras och hur projektet förankras i organisationen (Checkland & Holwell, 1998). ISD beskrivs som utvecklingen av specialanpassade system för en viss uppdragsgivare vilket är en komplex process som inkluderar förankringen i organisationen där implementeringen sker (ibid). I denna utvecklingsprocess så utförs oftast arbetet av antingen en för företaget intern IT-utvecklingsavdelning eller av konsulter som utför systemutveckling på kontraktbasis (Carmel & Sawyer, 1998).

De personer som arbetar med ISD, vilket kan beskrivas som en flerfaldig disciplin, måste behärska allt från de matematiska sidorna av programmering, till de sociala sidorna av organisationsutveckling och de psykologiska sidorna innefattade i användargränssnittet (Andersen, Kensig, Lassen, Lundin, Mathiassen, Munk-Madsen & Sødergaard, 1998).

Andra viktiga områden som måste behärskas är design, analys, företagspolitik, konflikthantering samt förmågan att kunna sätta sig in i andras arbetssituationer(ibid).



**Figur 3.3** Systemutvecklingens huvudelement (Andersen et al., 1998)

Den modell som visas i figur 3.3 är enligt Andersen et al. (1998) en generell bild av hur systemutveckling bedrivs. I modellen beskrivs hur systemutveckling kan delas in i två huvuddelar, produktinriktning och processinriktning, där den första fokuserar på *design*, *analys* och *realisering* medan den andra fokuserar på *planering*, *värdering* och *reglering*. Faktorerna *reflektion* och *förändring* påverkar både *produktinriktningen* och *processinriktningen*. (Andersen et al., 1998)

Det som i modellen beskrivs som visioner och existerande innebär att det är viktigt att reflektera över både existerande ting och de visioner som finns för ett projekt. De två inre momenten *utförande* och *ledning* beskriver processen för hur produkten tas fram samt hur projektledningen anpassas. Naturligtvis finns det ett beroende mellan *utförande* och *ledning* där det handlar om att reglera den ena efter den andra och tvärtom. (Andersen et al., 1998)

I konventionella ISD-projekt sker arbetet med inriktning mot utveckling för en specifik kund eller ett antal fördefinierade användare som oftast kan definieras tidigt i processen. Dessa användare är ofta inblandade i utvecklingen på ett flertal olika sätt och är ofta de som sätter upp kraven på vad systemet i slutändan ska klara av. Dessa projekt kan ofta finnas i organisationer med ganska stora resurser där det viktigaste med projektet är att det levererar en bra produkt och där

kostnaderna för utvecklingen är en sekundär faktor för projektet. För att bedöma hur lyckat ett konventionellt ISD projekt är används ett flertal mått förutom kostnader, som till exempel hur pass nöjda användarna är med systemet, hur väl det accepteras av användarna och kvalitén på systemet. (Carmel & Sawyer, 1998)

### 3.2.1 Metoder i praktiken

Vilka modeller olika utvecklare använder beror dels på vilka modeller de känner till och dels på ett antal andra faktorer. Hansen, Kautz & Jacobsen (2003) presenterar ett antal faktorer som påverkar det faktiska användandet av systemutvecklingsmetoder:

- Universality
- Confidence
- Experience
- Co-determination
- Introduction

Angående *universality* kan sägas att det är svårt att förändra och anpassa sig till nya metoder när det nuvarande systemet är konstruerat efter en specifik metod avvikande från den universella metoden vilket försvårar anpassningen. För att undvika denna problematik tillämpas i vissa fall prototyping för att göra metoden iterativ. (Hansen et al., 2003)

*Confidence* är en viktig faktor i utvecklingsarbetet, det är framför allt utvecklarna som måste känna att den metoden som används tillför något, driver arbetet framåt och producerar resultat. (Hansen et al., 2003)

*Experience* kan beskrivas som att använda sin kunskap inom ämnesområdet istället för att vägledas av det som metoderna föreskriver. Detta motverkar i sin tur involverandet av nya metoder som på detta sätt bortses ifrån. Detta kan då förklara på vilket sätt motarbetning av införandet av nya metoder sker. När sedan utvecklarna ställs inför nya och för dem outforskade situationer väljer de att lösa problemet med sina inarbetade metoder istället för att välja en metod anpassad för den aktuella situationen. (Hansen et al., 2003)

*Co-determination* är viktigt för att få med sig utvecklarna i utvecklingsprojektet. Det handlar om att få dem engagerade i sitt arbete och driva det framåt med nya idéer och förslag. Prestation och engagemang har här en tydlig koppling till varandra (Hansen et al., 2003).

*Introduction* kan beskrivas som hur utvecklarna introduceras för en utvecklingsmetod. Hur väl de blir insatta i metoden, utbildning av hur metoden fungerar samt att få testa hur den fungerar. Så när en använd metod förändras i



någon fas eller får ett helt nytt förhållningssätt till problemet som skall lösas är det viktigt att hela utvecklingsgruppen får förståelse för hur den påverkar deras arbete och arbetssätt. (Hansen et al., 2003)

Användandet av metoder i praktiken skiljer sig från hur de ursprungligen utformats (Hansen et al., 2003). Det är ofta så att utvecklare väljer delar av metoder för att använda i sitt eget utvecklingsarbete. Dessa delar kombineras sedan med andra verktyg för att passa väl in i den aktuella utvecklingssituationen och för att komma runt de begränsningar som metoderna har (Bansler & Bødker, 1993 enligt Hansen et al., 2003).

### **3.3 Standardsystem**

Standardsystem har i sig ingen direkt översättning till de mer vedertagna engelska termerna utan kan mest liknas vid COTS (commercial-off-the-shelf) eller Packaged Software. COTS system kan beskrivas som mjukvaruprodukter som är tillgängliga via mjukvarumarknaden till branscher med specialiserade behov och är oftast stand-alone applikationer (Vamanana, Wangb, Battab & Szczerbad, 2002). Packaged software kan enligt Carmel & Sawyer (1998) beskrivas som mjukvara som säljs som en produkt.

Ett standardsystem kan alltså sägas vara en färdig programvara som med viss anpassning av systemet kan utnyttjas för ett företags verksamhet, en sorts paketerad systemlösning. Standardsystemen kan ses som en generell systemlösning som är utvecklad av en leverantör för att passa ett flertal användares behov och är ett alternativ som finns för att företag ska slippa behöva uppfinna hjulet om och om igen (Nilsson, 1991). Dessa system har även utvecklats för att vara ett mindre riskfyllt alternativ till den konventionella utvecklingen av informations system som skötts genom till exempel egenutveckling av ett system från grunden eller genom att låta inhyrda konsulter utföra arbetet. Förutom att dessa system innebär en mindre risk för företagen så kan det ofta innebära ett billigare alternativ. Nackdelen med dessa system är dock att systemen inte alltid är så väl anpassade till företagets processer, utan antingen kan det krävas att organisationen anpassas till systemet eller att systemet anpassas efter organisationen (George, 2000). Standardsystemen omfattar en del av mjukvarumarknaden och har under senare år utvecklats till att bli en av de absolut största delarna på denna marknad. Men standardsystem är ingenting som är helt nytt, redan på 1950-talet såldes det MRP-system (Material Resource Planning) som var de första standard affärsapplikationerna. Dessa kom sedan att utvecklas till att omfatta andra delar av företagets processer och flera andra sorters affärsrelaterade standardsystem har uppkommit under tidens gång (Klaus, Roseman Gable, 2000).

Standardsystem är som tidigare nämnts en bred definition som omfattas av ett flertal olika sorters system. Inom detta fält kan sägas ingå flera olika sorters system, bland annat MRP, ERP (Enterprise Resource Planning), CRM (Customer Relation Management) och SCM-system (Suply Chain Management). Alla dessa är system som är skapade för att med någon form av anpassning kunna användas i

ett företags verksamhet, men det finns även andra system som säljs off-the-shelf som skulle kunna sägas ingå under termen standardsystem, såsom ordbehandlare, kalkyleringsprogram, spel med mera. De sistnämnda har dock valts att inte omfattas av definitionen av standardsystem i detta fall, trots att de är utvecklade för ett flertal användare, detta eftersom att de är utvecklade utan att ha affärsnytta som en grundtanke för systemen.

Som tidigare nämnts är standardsystem fortfarande inte ett helt problemfritt alternativ utan kan fortfarande skapa problem för de företag som väljer att köpa in dem. Implementeringen av standardsystem lider ofta av samma problem som konventionella systemutvecklingsprojekt, med förseningar och ökade kostnader till följd. Att införa ett sådant system innebär för företaget inte bara att företaget får in ny mjukvara som användarna måste anpassa sig till, utan även att en ny process för hur arbetet ska utföras måste införas. Det är viktigt i dessa lägen att företaget som implementerar systemet planerar för detta och även har den organisatoriska utvecklingen i åtanke vid implementeringsfasen. (Scott & Vessey, 2000)

Trots detta investerar många företag i inköp av standardsystem då dessa ofta leder till positiva förändringar i företagets infrastruktur och affärsprocesser, vilket i sin tur kan leda till omfattande besparingar och effektiviseringar av verksamheten (Scott & Vessey, 2000). Ett stort bidrag till organisationen som kan komma med ett standardsystem är en kunskapsförstärkning, då de flesta standardsystem är byggda på en stor kunskapsmassa som kommit av kontinuerliga uppdateringar av systemet, vilket kan leda till en mer professionell verksamhet (Nilsson, 1991). En annan av dessa förändringar som kan komma av ett införande av standardsystem är att företaget får en mer integrerad affärsverksamhet genom att ha integrerade informationssystem, vilket kan ge förbättringar inom till exempel kundservice. Systemintegration innebär en starkare länk mellan informationssystemen och databaserna och på så vis få möjligheter att få bästa möjliga grund för att analysera den data som finns i en organisation (Markus, 2000). Till exempel så kan en situation uppstå där arbete sker med ett flertal olika system, med ett flertal olika databaser där det måste hämtas information från av dessa för att kunna utföra arbetet. I ett väl integrerat system kan systemet utföra detta, medan det i arbete med ett dåligt integrerat system kan komma situationer där användaren måste arbeta med dubbla system och utföra jämförelsen manuellt.

### **3.4 Skillnader mellan ISD och standardsystemsutveckling**

Utvecklingsprocessen som ligger bakom ett standardsystem är, som tidigare nämnts, inte ett område som är särskilt väl dokumenterat, och väldigt lite forskning har gjorts inom området. Detta i stark kontrast till den konventionella systemutvecklingen (ISD, kap. 3.2) som har mognat under en lång tid och som numera har ett flertal väl utvecklade och testade metoder, vilket utvecklingen av standardsystem saknar. Istället har standardsystemsutvecklingen kommit att byggas runt löst dragna riktlinjer för hur arbetet ska utföras. (Carmel och Sawyer, 1998)

Litteratur  
- *Konsten att göra det specifika generellt* -

De dokumenterade skillnaderna mellan utveckling av standardsystem och konventionella system kan återfinnas på flera olika nivåer av utvecklingsarbetet. En av de främsta skillnaderna är standardsystemsutvecklingens brist på relationer till de slutgiltiga användarna av systemet. De typiska utvecklarna av ett konventionellt system arbetar antingen internt eller genom en konsultbyrå som utvecklar system på kontraktbasis, med en tydlig grupp av användare definierad. Medan det i ett företag som arbetar med standardsystemsutveckling inte arbetas mot en enda kund, utan mot en marknad (Carmel och Sawyer, 1998). Detta gör det väldigt svårt att ta fram användarkrav som lätt blir dåligt formulerade eller obefintliga (Carmel, 1993). I standardsystemsbranschen så är det främst företagets marknadsavdelning som kommer i kontakt med kunderna, vilket gör att utvecklarna blir beroende av marknadsavdelningen för att få feedback från användarna som kan användas för att göra krav för systemet (Carmel och Sawyer, 1998). Inte bara bristen på nuvarande användare kan komma att bli ett problem för företaget och utvecklarna utan även bristen på kontakt med framtida användare, vilket kan vara en kritisk framgångsfaktor för den framtida utvecklingen av systemet. Nu ska dock inte sägas att möjligheten för kommunikation med användarna via olika kanaler är omöjlig. Det finns ett antal möjliga metoder som kan användas, bland annat som tidigare nämnts är marknadsavdelningen en viktig kommunikationskanal, medan andra kanaler kan vara teknisk support, e-post, användargrupper, med mera (Carmel, 1993).

En annan sak som är specifikt för standardsystemsutvecklingen är de problem som kan uppstå vid utformandet av användarkrav. Även här utgör bristen på kända användare det stora problemet för utvecklarna. Att göra kravspecifikationen internt är en väg som kan fungera, men som skapar problem då det är svårt att avgöra huruvida den stämmer och är fullständig. Istället finns det ett antal tekniker som kan användas för detta såsom brainstorming, användarundersökningar, fokusgrupper, jämförelser med konkurrerande system och marknadsundersökningar. En dåligt definierad kravspecifikation kan leda till ett konstant tryck efter att ständigt uppdatera kravspecifikationen ytterligare, vilket kan ses som ett problem då kravspecifikationen förr eller senare måste låsas vid en slutgiltig version för att kunna skapa ett säkert system. Carmel (1993) menar att svårigheten ligger i att veta hur länge uppdateringen av kravspecifikationen kan fortgå och när det är dags att låsa sig.

Ett tredje utmärkande drag kan även sägas vara standardsystemets krav på sig att kunna användas på ett flertal olika plattformar då inte alla företag använder sig av liknande produkter. Detta ställer krav på utvecklarna att dels göra det möjligt för användning på ett flertal plattformar, men det innebär även ett mer omfattande testningsarbete. (Carmel, 1993)

Ytterligare något som är specifikt för standardsystemsutvecklingen är att utvecklingen måste anpassas efter marknaden och reagera snabbt när möjligheterna kommer. Detta innebär att arbetet måste kunna ske snabbt, samt att det måste finnas en möjlighet att kunna ändra och anpassa utvecklingen om det sker eventuella skiftningar i marknaden under utvecklingsprocessen. Detta gäller även uppdateringar av systemet som i de flesta fall måste följa på systemet i snabb

takt för att anpassa systemet till den rådande marknaden. Att det ser ut på detta sätt innebär också att produktens livscykel kommer att se annorlunda ut för de två olika sätten att utveckla. Konventionellt sker uppdateringar och felkorrigeringar kontinuerligt på det aktuella systemet. I standardsystemsutveckling arbetas precis som i den konventionella utvecklingen med uppdateringar och felkorrigeringar för det befintliga systemet. Vad som dock skiljer de båda åt är arbetet med nya versionsläpp av systemet, som kan innebära ganska så stora förändringar i grunden till systemet, vilket kommer diskuteras mer i kapitel 4.1. (Carmel, 1993)

Standardsystemsbranschen skiljer sig även åt i vad som fokuseras på att uppnå med systemet. De mått som appliceras på ett standardsystem är mått som vinst och effektivitet till skillnad från de konventionella systemen som värderas efter acceptans och belåtenhet hos användarna. Att standardsystemsutveckling är marknadsberoende innebär också att standardsystemen ligger under konstant press från marknaden. Så fort ett liknande system kommer ut som har implementerat en ny viktig funktion i systemet så måste denna funktion läggas till i nästa release i det egna systemet, oberoende hur långt processen har fortskridit, något som den konventionella systemutvecklingen inte behöver ta hänsyn till på samma sätt. Det kan också sägas att en skillnad mellan de båda är att de konventionella systemen svarar på någon form av krav, medan standardsystemen drivs av den teknologiska utvecklingen. (Carmel och Sawyer, 1998)

Standardsystemsutveckling är även en aning speciell i dess syn på implementeringsfasen då detta oftast är något som ligger utanför själva utvecklingsprocessen. I konventionell systemutveckling tas stor hänsyn till hur systemet ska introduceras och hur det ska skapas acceptans för systemet. Denna del är något som många av de företag som utvecklar standardsystem inte behöver bekymra sig om då denna tjänst kan överlåtas på en tredje part, till exempel en konsultfirma (Carmel och Sawyer, 1998).

### **3.5 Från litteratur till resultat**

Det har tidigare här i kapitel 3 presenterats information om både konventionell systemutveckling och det som funnits gällande utvecklingen av standardsystem. Det har framför allt presenterats två modeller som ligger till grund för fortsättningen av uppsatsen, vattenfallsmodellen och spiralmodellen. Ur dessa kan identifieras en mängd olika faser som de antingen har gemensamt eller som bara återfinns hos en av modellerna.

De faser som valts ut är, Förstudie, Planering, Utveckling, Testning, Implementering och Support. Att just dessa faser ansågs intressanta för studien grundades i att de har viktiga och kritiska roller i de modeller där de återfinns samt att de skulle täcka hela flödet även för standardsystemsutveckling. De andra valdes såldes bort på grund av att de inte skulle tillföra något, eller täcka in andra delar än de som täcks av de utvalda. De delar som inte innefattades av de utvalda faserna skulle inte gå förlorade då det från det kodade transkriptet klassificerades in i de utvalda faserna.

Litteratur  
*- Konsten att göra det specifika generellt -*

Det diskuteras i kapitel 3.1.3 hur vattenfallsmodellen och spiralmodellen inte uppfyller de krav som ställs på en utvecklingsmodell för standardsystem. Däremot tillkännages det att med vissa förändringar på respektive modell så skulle de passa för den undersökta formen av utveckling. Detta motiverar varför de utvalda faserna skulle täcka in hela spännvidden för standardsystemsutveckling. Vidare tas det upp, i kapitel 3.2, hur systemutveckling består av två huvuddelar, processinriktning och produktinriktning. Även dessa två delar anses täckas in av de identifierade faserna som då även inkluderar en mer generell bild av hur systemutveckling bedrivs

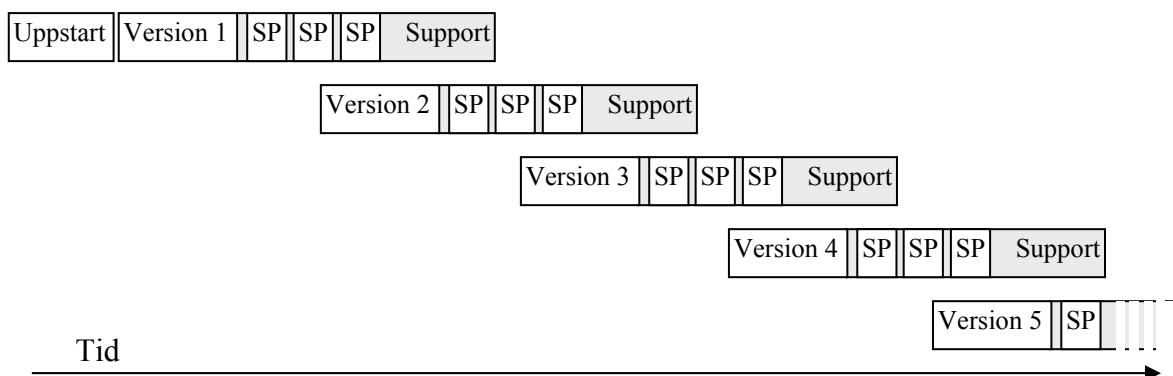
## 4 Resultat

I detta kapitel kommer att presenteras resultatet från studiens intervjuer. Uppdelningen är utförd efter resultatet från kodningen som genomförts. Först kommer mer övergripande information om produkterna och dess placering på marknaden. Senare i resultatdelen presenteras de olika stadierna eller faserna som identifierats i utvecklingsprocessen för standardsystem. Dessa är förstudie, planering, utveckling, testning, implementering och support.

### 4.1 Produktlivscykeln

Att tillverka standardsystem medför ett ständigt behov av underhåll och utveckling gällande produkten. Det har framkommit att uppdateringar och felkorrigeringar kan ges ut upp till en gång i veckan, vilket ställer stora krav på versionshantering och snabbt agerande för att åtgärda olika brister. Det som görs då kan vara mindre korrigeringar som bara berör en eller enstaka kunder. Dessa korrigeringar installeras då bara av de kunder som berörs men inkluderas och blir tillgänglig för alla kunder i nästa service pack.

Det har identifierats tre olika sorters uppdateringar av systemet, dels felkorrigeringar, dels service pack och dels nya releaser som ofta blir i form av en ny version och då innefattar större förändringar. Dessa tre tillsammans ger standardprodukten en kontinuerlig utveckling och ser till att kunderna får den specificerade funktionaliteten som garanterats.



Figur 4.1 Den uppfattade produktlivscykeln

Ovan (figur 4.1) illustreras hur olika versioner av produkten blir tillgängliga för kunderna, initialt sker en uppstartsfas där det definieras vilka funktioner som standardprodukten skall inneha. Efter det så sker utvecklingsprocessen för standardsystemets första version, det som i figuren ovan beskrivs som "Version 1". Själva utvecklingslivscykeln kommer vidare att beskrivas i kapitel 4.2. När denna är genomförd så blir standardsystemet tillgängligt för kunderna.

Resultat  
- *Konsten att göra det specifika generellt*

Då intar standardprodukten en annan ställning och supporten på produkten blir en allt större del, det är genom denna som förslag på förändringar samt felaktigheter i produkten kommer in. De nya kraven som ställts på produkten inkluderas, antingen som felkorrigeringar eller som service pack. Felkorrigeringar har inte tagits med i figuren på grund av den i förhållande lilla förändring som de ofta ger och på grund av att den inte alltid följer någon strukturerad process för att skapas eller berör så många kunder.

Det är även under ”Support” som planeringen av nästa version påbörjas. Vilket kan uttryckas som framkom under intervjun med en av respondenterna ”När man planerar att gå över från en version till nästa så kan man utgå från att den nuvarande versionen, plus de uppdateringar som lagts på som är en bra kravspecifikation för nästa generations system”. Men det är även andra saker som påverkar, till exempel önskemål som framkommit men som inte kunnat inkluderas i den nuvarande versionen av produkten och strategiska beslut i organisationen som tagits för produktens utveckling och spridning som beskrivs vidare i kapitel 4.1.1.

Som visas i figur 4.1 så överlappar versionerna, det vill säga att utvecklingsföretagen ger support även till tidigare versioner av produkten. Detta på grund av att det kan finnas organisationer som fortfarande kör gamla versioner av systemet. Att det är så har flera anledningar, det kan vara så att det inte har passat att göra en uppdatering i den aktuella organisationen på grund av den verksamhet som de bedriver eller att det ibland kan ha att göra med strategiska IT-beslut för organisationen, så att en implementering av en ny version sker först när verksamheten är redo för den. Andra faktorer som påverkar är att standardsystemets funktioner inte är det som är mest avancerat i organisationen utan att de kanske tillverkar en produkt som är väldigt avancerad och där standardsystemets funktion är att stödja den verksamheten, inte att försöka ta över den.

Den nya versionen som konstrueras får sedan en liknande versionscykel, med den skillnaden att det inte sker någon uppstartsfas utan det som skall konstrueras är mer klart än vad det är när en ny produkt skall tas fram.

Angående tidsskalan för varje version så är den en aning oklar då inga regler har framkommit under undersökningen, men vad som påträffats som vanligt är att det ofta släpps en ny version av standardsystemet i början på varje år. Det är också så att när systemet är nytt så släpps det nya versioner oftare beroende på att det som kunderna efterfrågar inte finns inkluderat i den dittills släppta versionen.

#### **4.1.1 Produktens spridning**

Det som har framkommit från studiens respondenter har varit att det finns två olika strategier för produktens spridning. Dels kan det vara att standardsystemet försöker inta en så stark ställning som möjligt inom ett smalare segment eller att det försöker fånga många kunder och täcka in större områden eller nischer. De undersökta standardsystemen har båda haft en vertikal utbredning, det vill säga att systemen varit inriktade på en smal nisch och specialiserat sig inom denna. Som

Resultat  
- *Konsten att göra det specifika generellt*

beskrivits i kapitel 2.2 så är denna undersökning gjord på två utvalda företag, Pipechain och Microcraft. Pipechain är ett logistiksystem som är avsett för större företag och Microcrafts produkt är ett mer komplett affärssystem som inriktar sig mot små och medelstora företag.

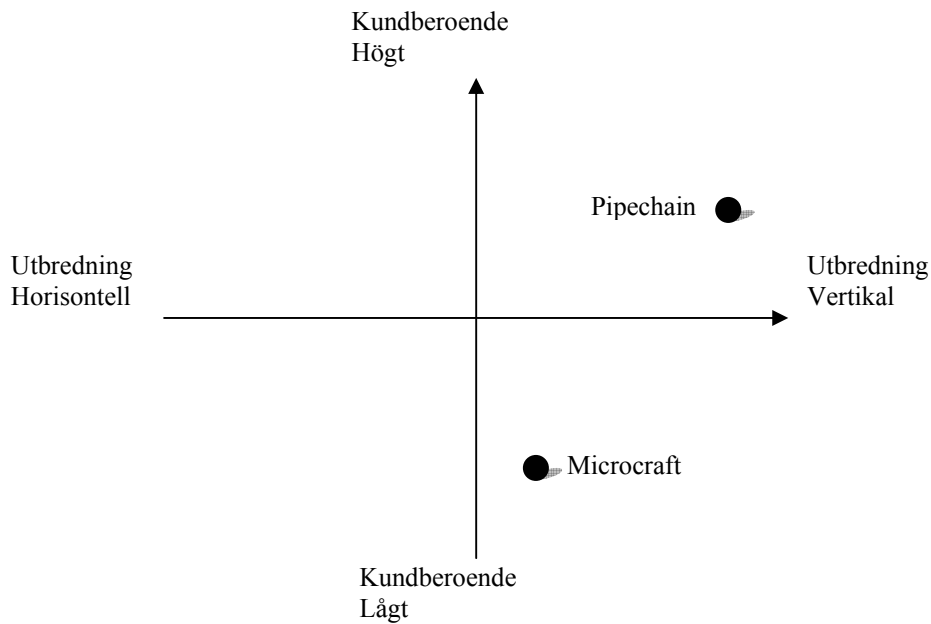
Fördelen med att ha en vertikal utbredning blir att möjligheterna att erövra kunder inom sin nisch blir större när det kan erbjuda dem en mer specialiserad produkt för det affärsområde som kunderna tillhör. Den stora nackdelen är att den totala kundbasen som standardsystemet attraherar minskar och det standardsystemsutvecklande företaget blir värre drabbat av andra aktörer som går in och tar marknadsandelar i just den branschen.

En annan faktor som kan vara relevant gällande produktens spridning är om standardsystemet riktar sig till små eller stora organisationer. Det kan vara lättare att sälja in ett system till en liten organisation, men den ekonomiska ersättningen blir ofta lägre vilket kan vara en nackdel. Säljs systemet in till en större organisation så är ofta den ekonomiska ersättningen god men det blir svårare att behålla standardfunktionaliteten i systemet och på så vis kunna sälja det även till andra. Detta beror på att påtryckningarna och beroendet till kunden blir mycket större i de fallen där det är en mindre utvecklare av standardsystem som säljer systemet till en stor kund.

Vad är då en stor kund? Det finns flera olika sätt att mäta vad en stor kund är, dels kan den omsättning som kunden har vara en mätare, dels kan antalet samtidiga användare vara en och dels kan det mätas i hur många transaktioner som utförs dagligen, det är svårt att säga vilken som är bäst. Men enligt studiens respondenter så är det dagliga antalet transaktioner den faktor som väger tyngst. Ovanstående resonemang ledde sedan fram till konstruerandet av figur 4.2 som beskriver hur studiens fallföretag befinner sig till varandra.



Resultat  
- *Konsten att göra det specifika generellt*



Figur 4.2 Standardsprodukters marknadsorientering

## 4.2 Utvecklingsprocessen

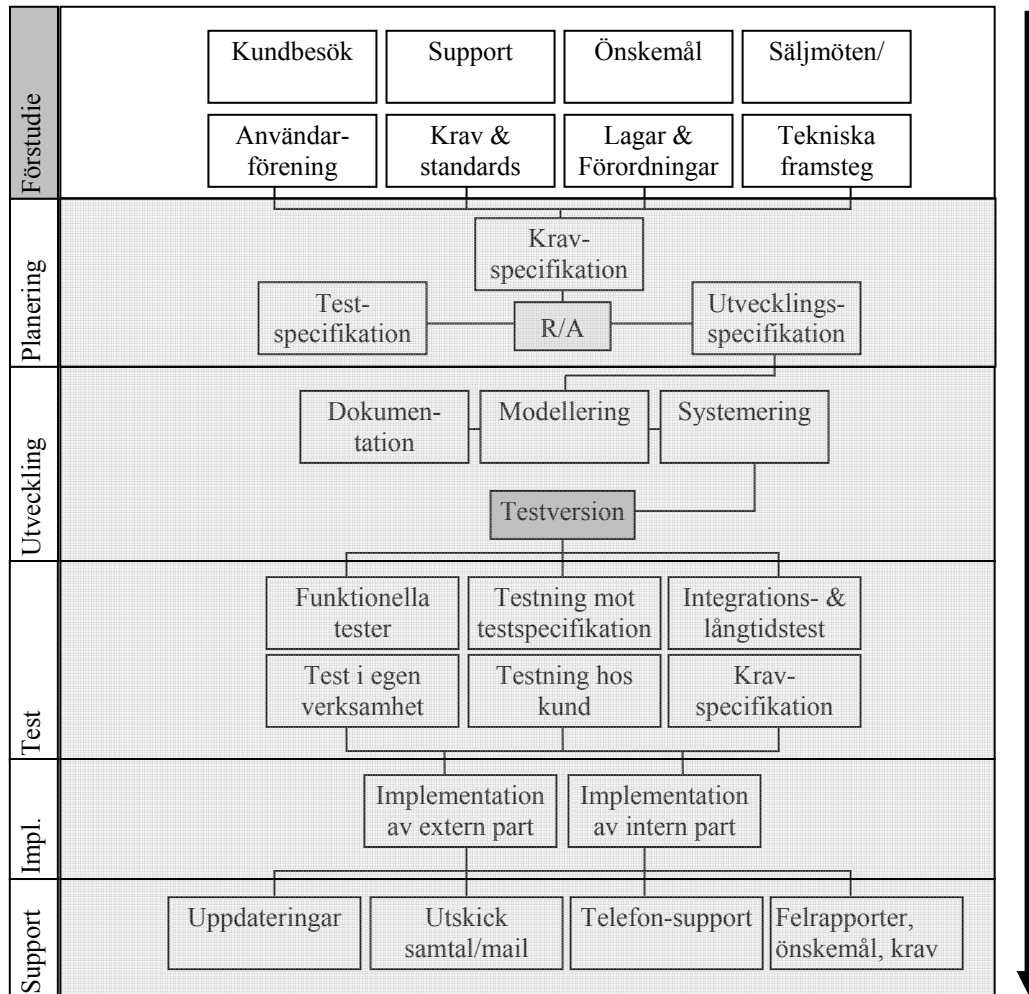
Det kommer här nu ingående att beskrivas hur utvecklingsprocessen för standardsystem ser ut i de fall som undersökts. Kapitlet är uppdelat efter de faserna som identifierades från litteraturen och som använts genomgående i uppsatsen. Faserna är Förstudie, Planering, Utveckling, Testning, Implementation och Support. Dessa faser kan närmast beskrivas som ett utvecklingsflöde där Förstudie är den första faser och Support den sista i en enskild utvecklingscykel som beskrivits i kapitel 4.1.

Naturligtvis innehåller varje fas en mängd olika steg där vissa är unika för standardsystemsutveckling och andra även återfinns i de utvecklingsmodeller som presenterats i kapitel 3. Alla identifierade steg har framkommit genom kodning och analys av studiens transkript som beskrivits i kapitel 2.4.1.

För att underlätta förståelsen för hur de olika delarna hänger ihop har det i figur 4.3 till figur 4.8 behållits även de delar som inte berörs av just den fasen som beskrivs. Figurerna nedan visar alltså inte hur iterationen mellan de olika delarna sker utan beskriver närmast ett idealflöde där ingenting blir fel eller måste ändras. Senare i figur 4.9 beskrivs däremot hur de olika delarna itererar.

Resultat  
- *Konsten att göra det specifika generellt*

### 4.2.1 Förstudie



**Figur 4.3,** Förstudie

Varje utvecklingsprojekt har en fas där projektet startas upp, oberoende på var produkten befinner sig i produktlivs-cykeln. Denna fas har valts att kallas förstudie då den är av undersökande karaktär och syftar till att bygga upp en förståelse för vad som senare ska göras i det aktuella projektet. Flera metoder finns för att samla ihop den informationen som behövs för att skapa sig en bredare förståelse över vad som skall göras. Beroende på om projektet befinner sig i uppstarten av ett nytt system eller om det befinner sig längre in i projektutvecklingsprocessen så kan metoderna som används skilja sig på ganska grundläggande basis.

När projektet befinner sig i uppstarten saknas antagligen en bredare kundbas vilket gör det svårt att definiera exakt vad som måste göras i systemet för att ett större antal organisationer ska kunna använda det. En förstudie till basen av systemet måste då istället läggas på en grundläggande nivå för att försöka förstå vad systemet ska ha för grundfunktion, för att kunderna överhuvudtaget ska vara intresserade av att i ett senare skede köpa systemet. Denna första del kan till

Resultat  
- *Konsten att göra det specifika generellt*

exempel baseras på en god affärsidé eller ett tidigare befintligt system som det finns potential i att utveckla vidare till ett standardsystem. En marknadsanalys eller liknande undersökning är i det här skedet något att föredra för att tillgodose sig information om hur marknaden ser ut.

När det i ett senare skede finns ett system färdigt och det har etablerats en kundbas finns det åtskilliga fler kanaler att använda sig av för att få kunskaper i vad som behövs utvecklas för att nuvarande kunder ska vara nöjda och för att kunna nå nya intressenter. De vanligaste metoderna för att samla in information från kunderna är:

- Användarförening
- Kundbesök
- Support
- Önskemål
- Säljmöten/systempresentationer

Men det finns även andra faktorer som påverkar vad som måste utvecklas utöver vad kunderna anser om systemet, dessa faktorer är:

- Tekniska framsteg
- Krav och standards
- Lagar och förordningar

I denna fas finns ett flertal saker som bör tänkas på. Svårigheterna med att veta vad kunderna verkligen tycker behöver utvecklas i systemet är den främsta av dessa. Det måste finnas en stor objektivitet och försök att få en bedömning utifrån den stora massan av kunder istället för att hela tiden utveckla utifrån den kunden som skriker högst. Det är också en viktig bit för systemet att här kunna samla in information med tillräckligt hög kvalitet från ett stort antal kunder då systemet annars riskerar att falla utanför ramen för ett standardsystem och börja utveckla ett system som blir alldeles för specifikt.

En annan viktig sak att tänka på är att inte outsourca alla sina funktioner som har med kundkontakt att göra. Som det framkom från en av studiens intervjuer: ”Ju fler filter man har desto mindre vet man om verkligheten”. Om det till exempel finns en outsourcad implementering så skapar det ju ytterligare ett filter då konsulterna som implementerar systemen blir ett extra filter för informationen att gå igenom innan den når utvecklarna. Vilket kan leda till att denna information hinner förvrängas på vägen.

#### **4.2.1.1 Användarförening**

En vanlig metod som används är en så kallad användarförening där representanter från de olika användarna samlas och diskuterar systemet inom ordnade former och där även företaget som utvecklar systemet kan ingå som en del av dagordningen genom att bidra med information om systemet samtidigt som det finns tillfälle att samla in information av vad användarna tycker om systemet. Detta ses som en bra kanal för diskussion mellan kunderna där utvecklarna kan se om olika förslag till förändringar i systemet stöds av den stora massan av kunder. Det är även ett tillfälle för kunder av olika storlekar att kunna framföra sina åsikter på en jämställd nivå, vilket givetvis är bra för mindre kunder vars förslag annars hade försvunnit bland mängden.

#### **4.2.1.2 Kundbesök**

Andra sorters kontakter med kunden kan även genomföras. Ett av det vanligaste sättet att samla in information om vad kunderna anser om systemet och hur vidare utveckling av detsamma kan ske, är genom att göra kundbesök. Kundbesöken kan vara organiserade som allmänna möten där folk från utvecklingsföretaget gör besök hos ett användarföretag för att studera hur systemet används och för att få kritik och idéer om vad som fungerar bra och vad som kan förbättras. Men besök hos kunder kan även vara i form av besök för att göra korrigeringar i programmet hos kunden, där en kontroll med kunden ofta kan göras för att höra hur de anser att systemet fungerar.

#### **4.2.1.3 Support**

När det finns ett färdigt system blir det viktigt för systemet att ha en fungerande support. Detta av ett flertal anledningar, vilket kommer diskuteras senare (se kap 4.2.6), men i denna fas främst för att det är en viktig kanal för kunderna att kunna uttrycka vad de anser om systemet. Ett vanligt sätt att samla in denna information är genom att samla in de åsikter och problem som supporten inte kan lösa i en åsiktsdatabas. I denna kan ses till exempel hur pass vanligt förekommande ett problem är och eventuellt också en prioritet över hur pass allvarliga problemen är för systemet. Detta har fördelen av att ge en sammantagen bild av den information som kommer in istället för att se den som enskild information från en specifik kund.

#### **4.2.1.4 Önskemål**

I många fall finns det kunder som har specifika önskemål som behövs tillgodoses för att de ska kunna utföra sitt arbete på ett mer effektivt sätt. Dessa önskemål kan komma in på ett flertal olika sätt, ett av dem som tidigare nämnts är support. Men även mer traditionella kommunikationskanaler används för att framföra önskemål, som telefonsamtal och e-post.

#### **4.2.1.5 Säljmöten och systempresentationer**

En annan viktig faktor för att ett standardsystem ska kunna växa sig större och bättre är att samla in information från de potentiella kunderna som finns till systemet. Ett arbete som givetvis är svårt då det inte med säkerhet kan sägas vilka

som kommer att bli systemets nya kunder. Detta kan göras genom att säljmöten och systempresentationer genomförs hos de organisationer som visat intresse för systemet. Vid dessa möten kan information framkomma om till exempel vilka delar av standardsystemet som är passande för dessa kunder och vad som kan komma att behöva utvecklas för att systemet ska kunna användas i dessa nya organisationer. Information från misslyckade säljförsök kan också vara intressant då en konkurrent fått sälja in sig istället. Detta genom att se vad de hade att erbjuda istället och om det på något vis går att använda sig av detta i sitt eget system.

#### **4.2.1.6 Tekniska framsteg**

De tekniska framstegen är något som även påverkar standardsystemsbranschen i allra högsta grad. Efterhand kommer det nya tekniker och nya protokoll som utvecklas till att bli standarder. På senare år har både J2EE och .NET utvecklats. Detta påverkar standardsystemsutvecklingen på så vis att det ofta skapar nya möjligheter för systemet, och eftersom detta är något som kräver stora resurser är det något som främst utvecklas på ett långsiktigt basis. Det är därför vanligast att det inte görs några uppdateringar för denna sortens utveckling till en redan befintlig version utan att det är någonting som kan komma med i nästa version av systemet som utvecklas.

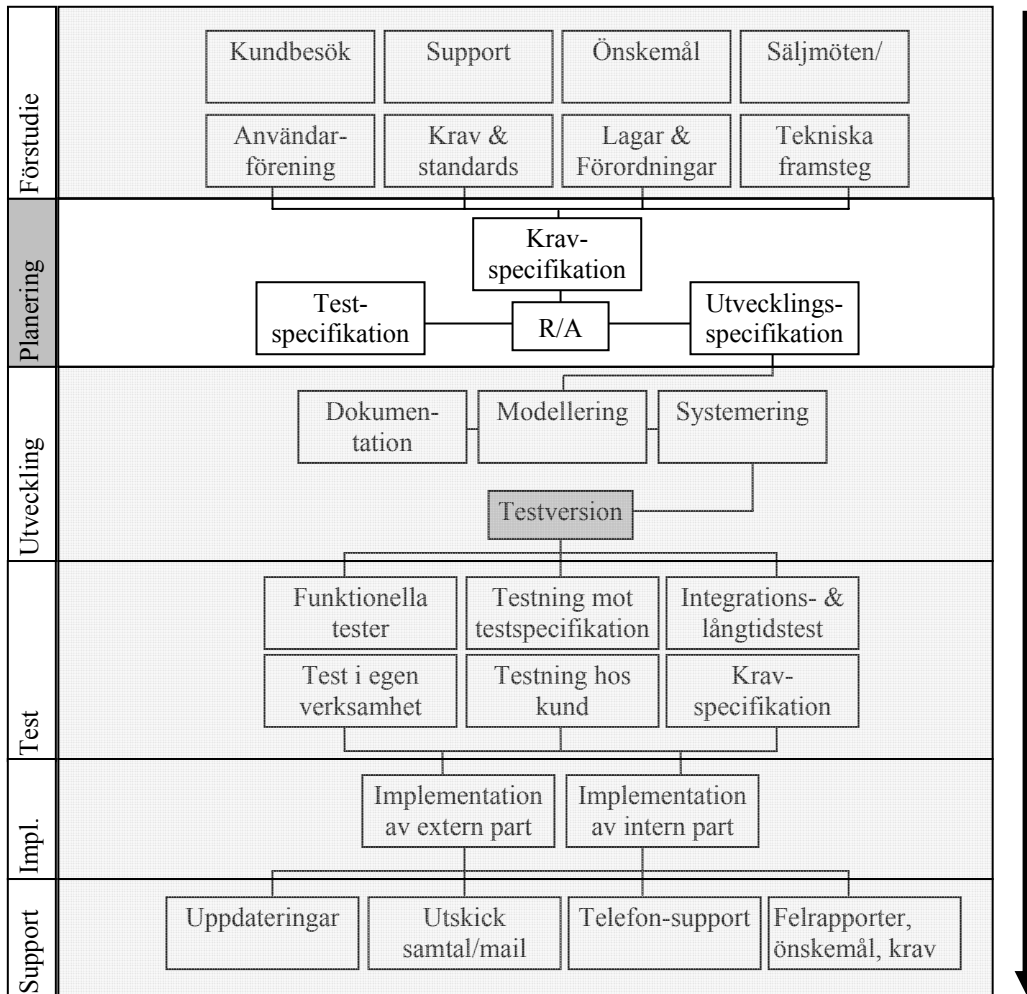
#### **4.2.1.7 Krav och standards**

Ibland kan det hända att nya krav och standards kommer från utomstående parter som måste tas hänsyn till i utvecklingen av ett standardsystem. Det kan vara att en bank ändrar sitt system för att sköta bankärenden via webben vilket innebär att om det standardsystem som utvecklats sköter detta så måste denna funktion utvecklas efter de nya standarderna. I dessa fall är det viktigt att som utvecklare bevaka och se vad som sker i den värld där systemet cirkulerar och försöka arbeta proaktivt. Då det kan underlätta att ha dessa ändringar planerade innan de realiserar av banken.

#### **4.2.1.8 Lagar och förordningar**

Lagar och förordningar är viktigt att ha med i sina beräkningar som standardsystemsutvecklare eftersom detta är någonting som kan komma att påverka utvecklingen i produkten. Detta är även någonting som är svårt att själv påverka utan är en del som måste accepteras som det är. Detta kan röra sig om att en kund till exempel vill etablera försäljning i ett annat land med andra lagar och regler än i det där de befinner sig, vilket då skapar nya krav på systemet. Ändras en lag som på något vis kan komma att påverka någon kund av systemet måste även systemet anpassas så att det uppfyller dessa nya krav. Även här är det viktigt att studera trender och se vad som är på väg att hända inom de världar som systemet kan tänkas påverkas av.

## 4.2.2 Planering



**Figur 4.4,** Planering

Planeringsfasen följer direkt efter förstudien och är till för att värdera och utveckla det som kommit fram i förstudie fasen. Det är en fas som i mångt och mycket handlar om att förhandla och att försöka hitta möjligheter för att kunna utveckla de delar som framgått att de behöver utvecklas. I stort kan sägas att fasen består av fyra olika delar:

- Kravspecifikation
- Risk analys (R/A i modellen)
- Testspecifikation
- Utvecklingspecifikation

#### **4.2.2.1 Kravspecifikation**

Den första delen som görs är en kravspecifikation som bestämmer i detalj vad systemet ska göra för kunderna. Detta blir till stor del en specifikation som är gjord för att kunna kommuniceras och måste därför vara gjord på ett sätt så att säljare, kunder och utvecklare kan förstå vad som efterfrågas. Genom att ha en sådan specifikation kan på ett bra sätt garanteras att utvecklare och kund har liknande uppfattningar om vad som ska utvecklas.

En kravspecifikation är baserad på information från en eller ett flertal av kanalerna i förstudie fasen. Men utvecklas en ny version av systemet kan det även sägas att den gamla versionen är en bra grund för en kravspecifikation, då det nya systemet måste klara av att göra det som det gamla klarar av. Denna specifikation är en officiell bild av vad som ska utvecklas och blir alltså en övergripande bild för vad kunderna vill att ha utvecklat i systemet. Det är här viktigt att göra bedömningar utifrån vad som är viktigt för kunden, men även vad som är viktigt för systemet. De önskemål som kunderna har kanske leder till att systemet passar ett visst antal kunder bättre, men det kan också innebära att systemet blir spretigt och försöker täcka in fler områden än vad det är tänkt att det ska göra.

Att det en gång i början av utvecklingsprocessen har gjorts en kravspecifikation behöver heller inte betyda att dessa funktioner är ristade i sten, utan är en pågående process, vilket framkom under en intervju: ”En specifikation är inte färdig bara för att man lämnar ifrån sig ett dokument. Det kommer upp nya saker under arbetets gång, saker läggs till och saker tas bort”. Det kan hända att nya saker framkommer under utvecklings- (se kapitel 4.2.3) eller testningsfasen (se kapitel 6.2.4) och på så vis gör att det blir tvunget att gå tillbaka till kravspecifikationen för att göra ändringar i densamma.

#### **4.2.2.2 Riskanalys**

Riskanalysen görs för att avgöra om någonting är värt att utveckla eller inte, eller till och med i vissa fall om det överhuvudtaget är möjligt att utveckla. Att göra kalkyleringar för vad som kommer att vara lönsamt att utveckla är ganska så svårt då det aldrig med säkerhet kan sägas hur lång tid som kommer krävas och vilka resurser utvecklingen behöver ta i anspråk. Därför byggs en sådan analys kring en hel del spekulationer. Men som det framkom under studiens intervjuer så finns det även andra sätt: ”Pengen är ett ganska bra mätinstrument. Det finns många önskemål men när det kommer till hur mycket man är bered att betala, vill du ge tiotusen för den här funktionen? Nej... då är det antagligen inte så viktigt för deras verksamhet.”

Att på så vis kunna göra en avvägning som är baserad på om någon är beredd att betala för det som utvecklas eller inte är en vanligt förekommande metod. Kan det redan innan utvecklingen börjat på något sätt säkerställas ekonomin för en viss utvecklingsdel så är den lönsam att utveckla i de flesta fall. Men ibland kan det uppstå en situation där frågan kommer upp om en funktion som ska utvecklas är viktig för systemet eller ej, oavsett om ekonomin för den finns eller ej. Vissa funktioner kan vara bra att ha, men kan ligga lite före sin tid så att den kommer att

ge mer avkastning längre fram i tiden och vara viktiga för systemet av den anledningen. Andra funktioner kan vara ekonomiskt lönsamma på kort sikt då ett antal företag vill ha den och är villiga att betala för den, men samtidigt kan den få systemet att gå utanför sitt målområde. Något som kan innebära att det blir svårare att sälja i ett framtidsperspektiv.

#### **4.2.2.3 Testspecifikation**

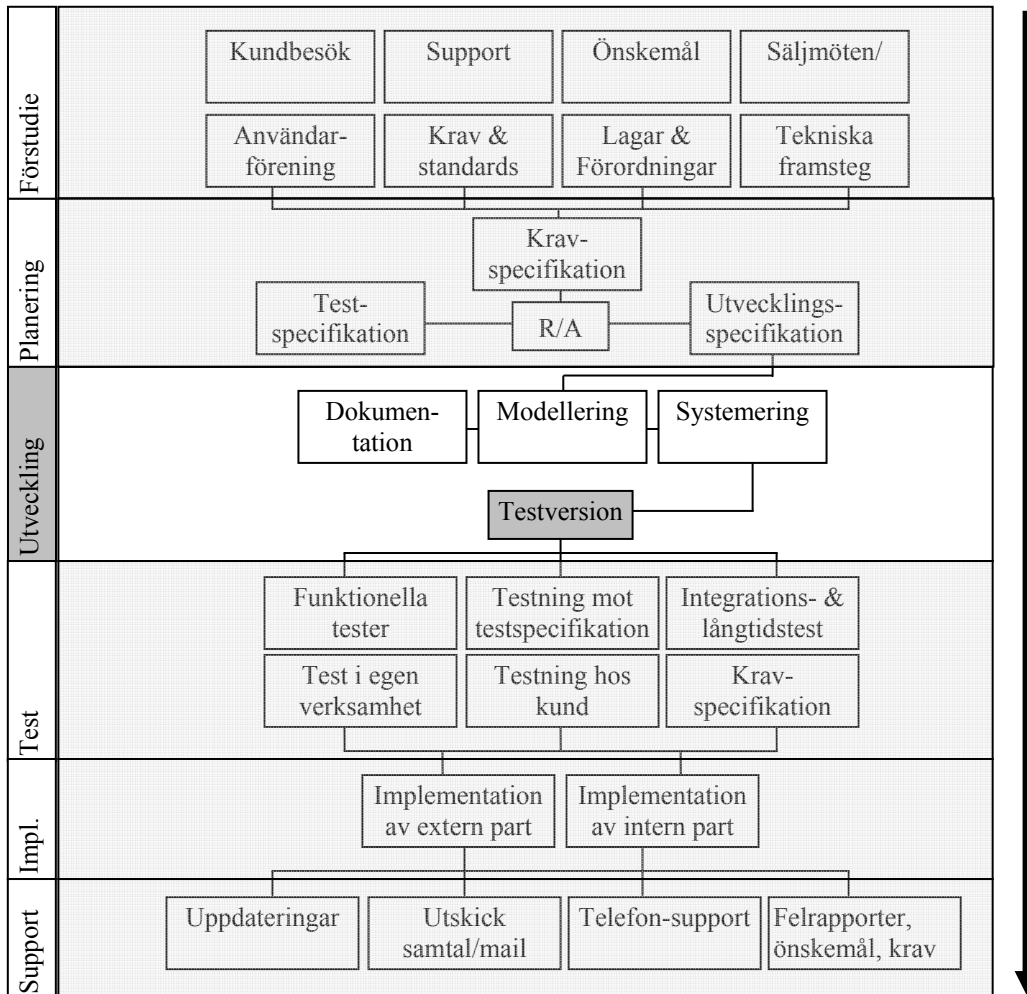
För att säkerställa att testningen görs på ett bra sätt så utvecklas det i ett tidigt stadium användarfall. Testspecifikationen görs redan innan utvecklingen är påbörjad eftersom att testfallen inte ska påverkas av produkten, utan byggas på vad kunden fastiskt är intresserad av att få. Risken finns att det i fall där utvecklaren skapar testfall efter det att produkten är färdig får testfall som är anpassade till att fungera som utvecklaren har skapat systemet istället för att testfallen är gjorda efter hur kunden ville att det skulle fungera i första hand. Detta har att göra med att utvecklarna lätt kan bli hemmablinda och anse att eftersom det är skapat på detta sättet de har gjort, så ska systemet fungera på det sättet, istället för att tänka på vem som egentligen är användaren. Testfallen kommer sedan att användas i testfasen som kommer i ett senare skede av utvecklingsprocessen (se kapitel 4.2.4).

#### **4.2.2.4 Utvecklingsspecifikation**

Utvecklingsspecifikationen är en utökning av den tidigare gjorde kravspecifikationen (se kapitel 4.2.2.1). Till skillnad mot kravspecifikationen är denna dock inte gjord för att kunna diskuteras gentemot kund, utan den är snarare gjord för att utvecklarna ska kunna komma vidare i sitt arbete. Kravspecifikationen kan i många fall vara skriven på ett för sammanfattande och övergripande sätt för att användas som specifikation för utvecklarna. Istället görs en mer detaljerad teknisk förklaring, en beskrivning av vad som ska utvecklas som kunden antagligen inte hade haft någon nytta av att se, eller som kunden kanske inte ens hade begripit sig på. Med hjälp av denna utvecklingsspecifikation kan sedan arbetet gå vidare och ett underlag för att kunna planera det kommande utvecklingsarbetet är gjort. Inte heller denna specifikation är helt klar förrän i slutet av alla tester har gått igenom med positivt utfall. Görs det ändringar i kravspecifikationen är det även troligt att det behöver göras ändringar i utvecklingsspecifikationen.



### 4.2.3 Utveckling



**Figur 4.5,** Utveckling

I denna fas sker den tekniska utvecklingen av det som är tänkt att utvecklas. Detta är en fas som kännetecknas av iteration och där mindre steg bakåt in i planeringsfasen och framåt in i testningsfasen hör till vanligheterna. Denna fas är uppdelad i främst tre stycken stora delar:

- Modellering
- Dokumentation
- Systemering

Det finns ibland risker i denna fas då utvecklingen kan gå för sakta, vilket kan innebära att marknaden hinner förändras vilket leder till att utvecklingen måste ändras. Då standardssystemen styrs i allra högst grad av marknaden kan en trög

utvecklingsfas även innebära att systemet kommer för sent ut på marknaden, vilket kan innebära att de fördelar som systemet skulle ha mot marknaden uteblir.

#### **4.2.3.1 Modellering**

Modelleringen är en viktig del av utvecklingsarbetet eftersom det underlättar mycket av det arbete som ska göras och det hjälper även utvecklare att på ett bättre sätt utföra sina arbetsuppgifter. Modelleringen kan bestå av verksamhetsmodeller, tekniska modeller och annat liknande arbete som bör göras innan kodningen påbörjas. Modelleringen är, visade det sig, dock inte enbart något som sker strikt före kodningen och som sedan inte rörs igen utan detta är en process som är konstant återkommande under utvecklingsfasen, vilket detta citat från en av respondenterna tyder på: "... det är ju alltid fråga om att tänka först och utföra sen. Men och andra sidan så tänker man och så utför, och så tänker man om och utför om, tills allting kommer samman."

#### **4.2.3.2 Dokumentation**

Dokumentationen är en viktig del av utvecklingsarbetet då detta kan vara en stor resurs för att senare göra ändringar inom det som gjorts och få en förståelse för hur arbetet utförts, samt för att få en överblick över systemet.

Teknisk dokumentation är en sorts dokumentation som är ganska så viktig. Den beskriver i stora drag hur systemet är uppbyggt och kan även innehålla delar som systemets olika användningsfall där det kan ses hur olika transaktioner går till i systemet och vad som händer om till exempel en viss knapp trycks ned. Men dokumentation kan även användas i form av regelverk eller kokböcker för hur kod ska skrivas, detta för att se till att ha en gemensam standard genom hela systemet.

#### **4.2.3.3 Systemering**

Denna del innefattar dels kodningen av systemet men även utformandet av detsamma. När ett standardssystem ska kodas är det viktigaste att det finns en plattform till systemet som är möjlig att göra förändringar i. Det vanligaste är att det arbetas med en skiktad lösning där det finns en kärna i systemet där inte särskilt många ändringar utförs mellan de olika versionssläppen, utan att ändringarna i applikationen kan utföras i ett annat skikt. På så vis kan systemet utvecklas snabbare och med mindre besvär och det gör även att det är lättare att göra förändringar och justeringar på systemet då det skulle komma att behövas. På detta vis bevaras även kärnan av systemet intakt och felfri vilket leder till att mängden allvarliga fel kan minimeras.

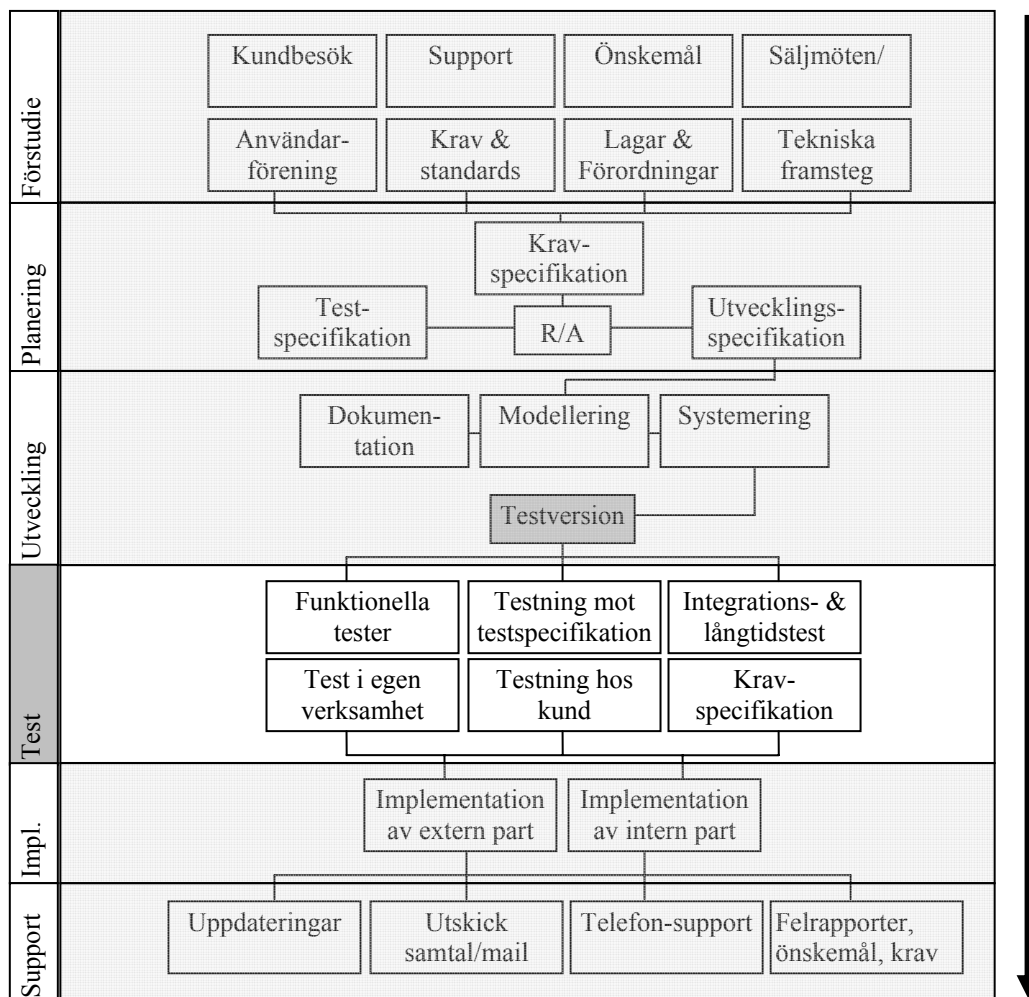
Utformningen av ett systems användargränssnitt är också en viktig del då det måste tänkas på att det utvecklas ett system för ett flertal användare utan att veta exakt vilka det är som kommer att använda det. Det vanligaste sättet att arbeta med detta är att försöka följa redan satta standarder som de flesta som använder sig av systemet redan är vana vid, till exempel Windowsstandarder. Men det är även viktigt att det finns ett flertal olika möjligheter som användare då alla inte ligger på samma nivå, vilket till exempel kan göras genom att utveckla

Resultat  
- *Konsten att göra det specifika generellt*

gränssnittet för att vara enkelt och lätt att använda, men samtidigt ha funktioner som gör att mer erfarna användare kan ta sig snabbt fram genom systemet.

Det är i denna fas inget ovanligt att tillfälligt gå in i andra faser. Till exempel om det skulle upptäckas att det här på ett lätt sätt skulle kunna täcka in ytterligare en sak som det tidigare inte tänkts på kan tas med i utvecklingen. Då kan projektet gå ett steg tillbaka för att göra förändringar i krav-, utvecklings- och testspecifikation för att sedan justera modelleringen så att kodningen kan fortsätta. Det händer även att det tas en steg in i testfasen (se kapitel 4.2.4) för att få snabb feedback på att rätt saker har utvecklats.

### 4.2.4 Testning



Figur 4.6, Testning

Testningen är en fas som kan avsluta den iterativa processen som beskrivs i figur 4.9. Detta är en viktig fas som skiljer sig från konventionell systemutveckling på grund av att det finns flera parter inblandade, som kan bli drabbade om deras

system plötsligt slutar att fungera på grund av en förändring i standardprodukten. Därav återspeglas det även i omfattningen av denna fas som har framställts som mycket viktig för utvecklingen av standardsystem i de fall som undersökts. Sex olika sorters tester har visat sig i undersökningen:

- Testning mot testspecifikation
- Testning mot kund
- Integrationstest
- Funktionella tester
- Test i egen verksamhet
- Test mot kravspecifikation

#### **4.2.4.1 Testning mot testspecifikation**

Testningen mot testspecifikation är en viktig del av utvecklingen av standardsystem. Själva testspecifikationen konstrueras under planeringsfasen, innan utvecklingen påbörjas, som återfinns i figur 4.9 och inleder den iterativa processen. Testspecifikationen består av ett antal testfall som är konstruerade utifrån en viss miljö, ibland ett specifikt företags miljö om det är så att funktioner som har adderats kan ha påverkat hur just detta företag använder produkten. Det är så att denna del är den viktigaste faktorn vid en värdering av de test som utförs på standardsystemet innan det blir tillgängligt för alla kunder. Omfattningen av denna test inkluderar mer än en enkel funktion och ofta större delen av produkten beroende på hur väl specificerade testfallen är. Ofta är det så att dessa tester resulterar i att utvecklingscykeln itererar, antingen tillbaks till utvecklingsfasen eller till planeringsfasen, detta beskrivet i figur 4.9.

#### **4.2.4.2 Testning hos kund**

Testningen hos kund är till för att testa så att produkten fungerar i kundernas miljö. Vilka kunder det blir bestäms till stor del efter vilka kunder det är som beställt eller önskat funktionen eller funktionerna som har utökat produkten. Men även kunder som kan tänkas ha berörts av att funktionen eller funktionerna har införts. För att genomföra denna testning på ett enkelt sätt används ofta Alfa- och Betaversioner av standardsystemet som kunderna får använda och som sedan utvärderas. Denna typ av testning sker i de fall som undersökts alltid när det sker en större förändring i standardsystemet, men däremot förekommer den inte alls när det bara är små korrigeringar som genomförts. Det är väldigt viktigt att inte denna test stör testföretagets verksamhet, därför sätter företaget själva eller i samarbete med standardsystemsutvecklaren upp en testmiljö på företaget som används parallellt med den ordinarie programvaran och som sedan kontrolleras ha utfört det som avsågs att utföra.

#### **4.2.4.3 Integrationstest**

Denna fas används för att testa både enskilda funktioner och hur dessa funktioner interagerar med övriga delar av produkten. Ofta kan någon del av produkten som är centralt belägen ställas in så att den automatiskt kommunicerar med sina närliggande delar och skickar information fram och tillbaka under en längre tid. Därefter så kontrolleras det så att rätt saker har utförts av standardsystemet.

#### **4.2.4.4 Funktionella tester**

Under faser funktionella tester har sammanslagits tester som har med det funktionella användandet att göra. Fyra olika slags tester har identifierats, plattformstest, prestandatest, kommunikationstest och användartest. Plattformstestetets funktion är att testa den plattformen som standardsystemet bygger på och hur funktionerna reagerar i den aktuella miljön. Prestandatest kontrollerar mer hur systemet hanterar resurser, det kan vara att programmet inte upptar mer och mer internminne och på så vis medför en större och större belastning på serversidan, vilket i slutändan innebär att systemet går ner och måste startas om.

För de delar av ett standardsystem som kan komma att kommunicera över Internet görs mycket ofta rena kommunikationstester för att kontrollera så att systemet skickar och tar emot den efterfrågade informationen korrekt. Användartest används ofta för stora förändringar och i mån av tid, det har inte framkommit att detta skulle vara en viktig del att testa, men den finns hela tiden i åtanke. För de standardsystem som undersökts har det varit gemensamt att användarna involverades i uppstartsfasen och i konstruktionen av den första versionen av standardsystemet (se figur 4.1).

#### **4.2.4.5 Test i egen verksamhet**

Vissa standardsystem har den fördelen att det går att testa dem i den egna miljön (på utvecklingsföretagen). Det är sällan så att det går att köra hela standardsystemet, men de delar som fungerar till detta används för att testa produkten i den egna verksamheten. Detta kan vara ett sätt att testa produkten i ett tidigt skede och kunna åtgärda brister i systemet innan det når någon kund. Det ger även medarbetarna, som inte arbetar med utvecklingen, en insikt i produkten och dess utveckling som är svår att uppnå annars.

#### **4.2.4.6 Test mot kravspecifikation**

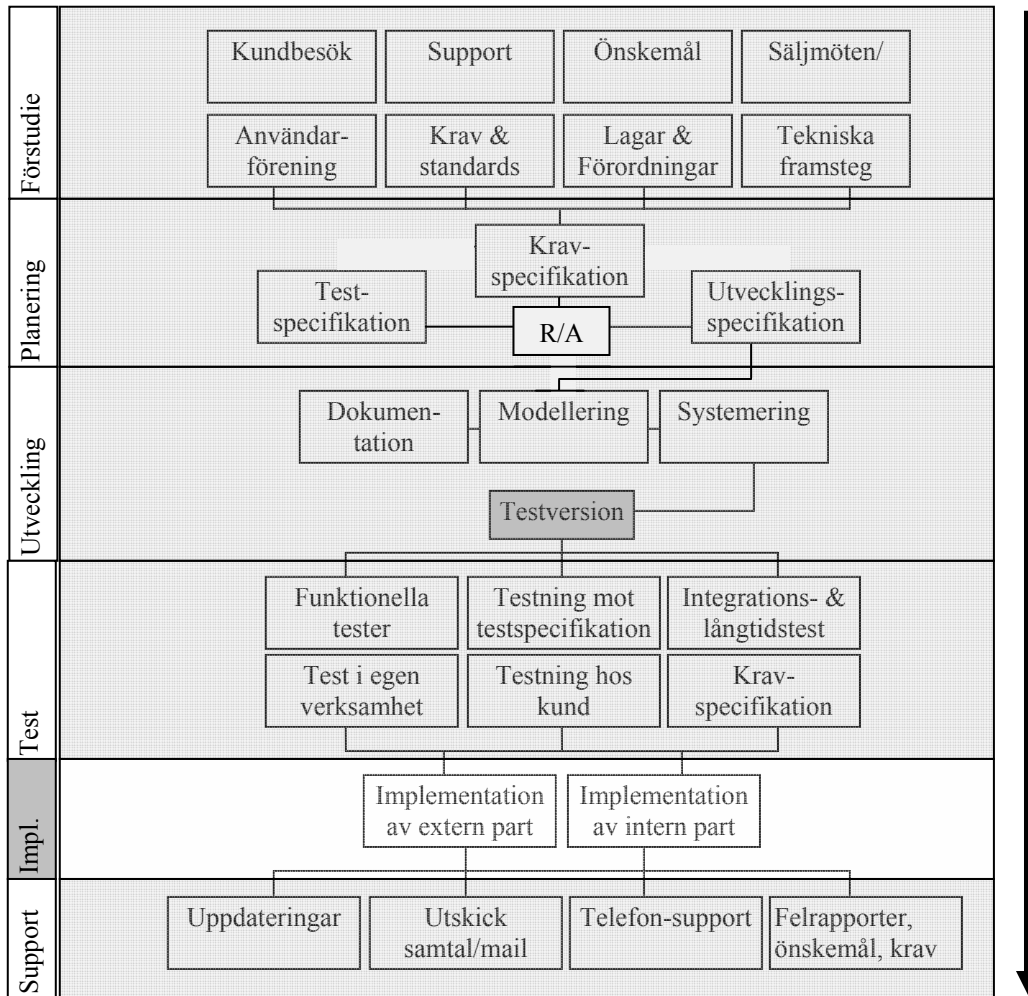
Testningen mot kravspecifikation är en viktig del som kan återfinnas både i konventionell systemutveckling och i utvecklingen av standardsystem. Denna del kompletterar testningen mot testspecifikation, men är mer koncentrerad på en viss del. Ofta är det så att kravspecifikationen är avsedd för antingen förändring av en viss funktion eller nykonstruerandet av den. Men det kan även vara konstruerandet av en ny version av standardsystemet.

Ofta ligger den innan ”Test mot testspecifikation” och kontrollerar att alla delar och tekniska specifikationer har uppnåtts. Just denna test genomförs nästan uteslutande av produktchef eller projektledare som har en övergripande syn på

Resultat  
- *Konsten att göra det specifika generellt*

standardsystemet. Detta för att det krävs erfarenhet för att se hur olika delar interagerar i ett större system, vilket togs upp som en viktig egenskap för att kunna göra relevanta tester generellt.

### 4.2.5 Implementering



**Figur 4.7,** Implementering

Det har framkommit skillnader mellan de undersökta fallen gällande hur de implementerar sina standardsystem hos kunderna. Ett av företagen hade en nära samarbetspartner som skötte större delen av deras implementationer och utvecklingsföretaget fungerade då istället som en resurs för dem, vilket kommer att beskrivas mer ingående i kapitel 4.2.5.1.

Det andra företaget som har undersökts skötte själva implementationen av sitt standardsystem, vilket beskrivs i kapitel 4.2.5.2. På grund av detta har det bara kunnat presenteras två från varandra skilda situationer:

- Implementation av extern part
- Implementation av intern part

#### **4.2.5.1 Implementation av extern part**

I fallet när det var en extern part som implementerade systemet så fungerade detta företag som en kund för utvecklingsföretaget, det var de som köpte standardsystemet och som sedan gjorde anpassningarna på det vid implementeringen. På detta vis kunde utvecklingsföretaget inrikta sig på att bygga standardfunktionaliteten och den externa parten kunde fokusera på att göra implementeringar.

Därefter så var det utvecklingsföretaget som stod för underhåll och uppdatering av standardsystemet och den externa parten som såg till att anpassningarna som var gjorda på systemet inte förändrades. Det framkom då att detta var en stor fördel som gjorde att fokus i organisationen kunde läggas på det området som det enskilda företaget är bäst på.

De delar som har identifierats och som finns beskrivna nedan i kapitel 4.2.5.2, ”Implementation av intern part”, Integration mot andra system, Konfiguration av systemet och Tilläggsfunktioner är även dem tillämpbara på den sekundära beskrivningen som givits av hur implementationen av den externa parten gick till.

#### **4.2.5.2 Implementation av intern part**

Den stora skillnaden mellan implementation av intern part i jämförelse mot implementationen av den externa parten är att det finns en djupare direkt kunskap om vad som är möjligt att göra med systemet. Det som gör det möjligt att anpassa standardsystemet är till stor del inställningar av parametrar för systemet tillsammans med formatkonverteringar när variabler skickas in och ut ur systemet.

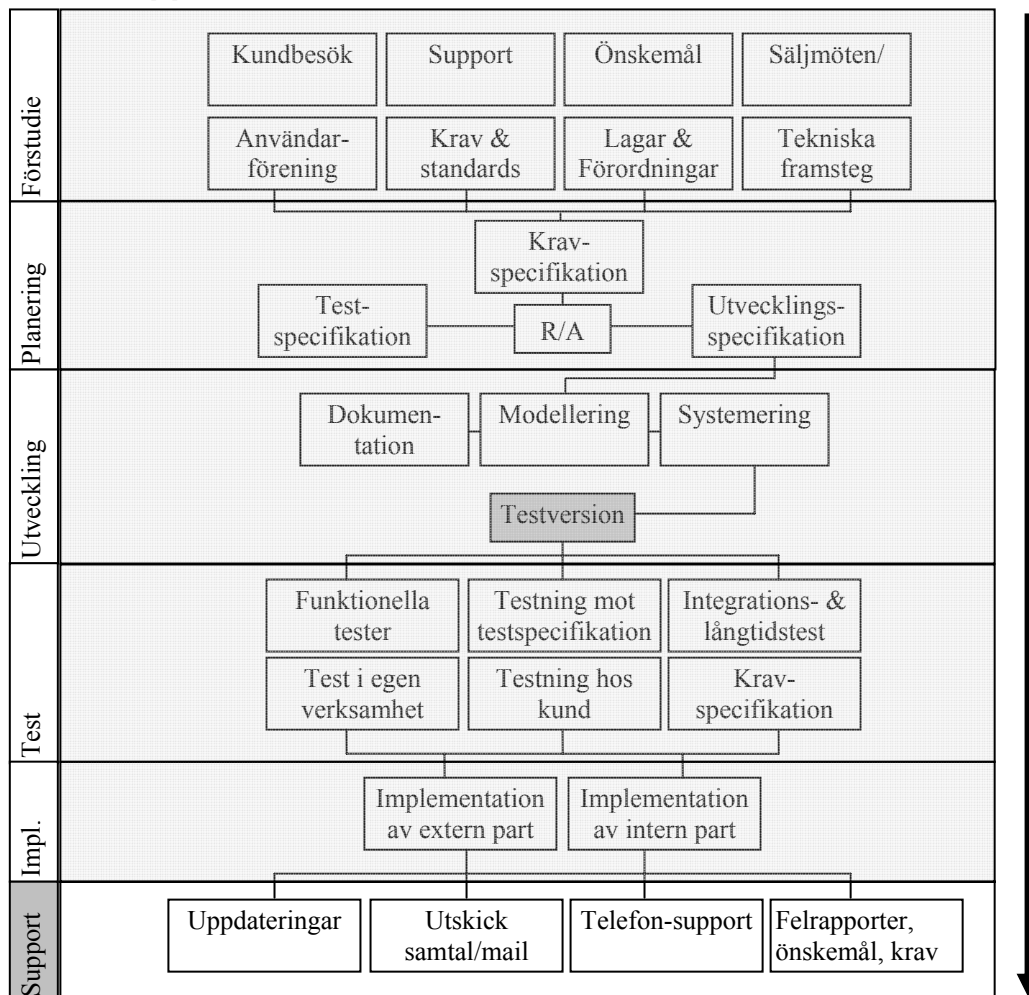
Just integrationen mot andra system är en viktig del, det kan dels vara konventionella system som skall integreras mot och dels kan det vara andra standardsystem som det skall integreras mot. Det som skiljer dessa två är att standardsystemet förändras oftare än det konventionella systemet och kräver då mer underhåll enligt det fall som undersökts. Det kan vara svårt att avgöra hur dessa kopplingar skall ske och vilka tekniker som skall användas särskilt när det gäller verksamhetskritiska flöden.

Olika standardsystem behöver olika mycket konfiguration för att fungera i olika organisationer, det handlar om att anpassa standardsystemet till just den verksamhet det skall operera i. Generellt kan sägas att ju mer horisontell utbredning som standardsystemet har enligt figur 4.2, desto fler variabler är det som måste ställas innan systemet fungerar enligt studiens respondenter.

Resultat  
- *Konsten att göra det specifika generellt*

Olika standardsystem täcker in olika mycket av en organisations verksamhet, det kan därför vara upphov till att det måste skapas mindre tilläggsfunktioner som stöttar standardsystemet mot verksamheten. Studiens respondenter har haft delade meningar om hur dessa tilläggsfunktioner sedan skall underhållas, dels fanns uppfattningen att dessa funktioner skulle underhållas skilt från standardsystemet och dels fanns uppfattningen att funktionerna skulle underhållas tillsammans med standardsystemet. De som tyckte att tilläggsfunktionerna skulle underhållas tillsammans med standardsystemet var framför allt utvecklarna, medan det hos produktcheferna fanns en uppfattning att de skulle hållas utanför standardsystemet och underhållas separat om det nu var nödvändigt.

### 4.2.6 Support



**Figur 4.8,** Support

Support är en del som också återfinns i konventionell systemutveckling, men för standardsystem blir den mer komplex och mer omfattande när det gäller att ta in intryck från fler parter. De olika sorters support som framkommit av undersökningen är:



- Uppdateringar
- Utskick samtal/mail
- Telefon-support
- Felrapporter, önskemål, krav

#### **4.2.6.1 Uppdateringar**

Standardsystem är i kontinuerligt behov av olika former av uppdateringar, det har identifierats tre olika former som varit förekommande i de båda fall som undersökts. Dels är det rena felkorrigeringar som kan behövas för att systemet skall utföra vad det är sagt att utföra och inte något annat. Dels kan det vara service pack som innehåller lite större förändringar och som både kan vara en samling av de senaste felkorrigeringarna men även tillföra systemet nya funktioner som blir tillgängliga för alla. Dels kan det vara nya releaser av standardsystemet där systemet går över från en version till en annan och innehållande större förändringar som även kan ha påverkat interaktionen inom produkten.

Det som ligger bakom dessa uppdateringar är förutom krav från den aktuella kunden även påtryckningar som kommer från andra håll. Det kan vara både från detta företags kunder likaväl som det kommer krav från myndigheter och lagstadgade krav.

#### **4.2.6.2 Informationsutskick**

Supporten har även en funktion att tillhandahålla information till utvecklingsföretagets kunder. Dessa informationsutskick kan gälla allt från nya funktioner till olika typer av undersökningar. Framför allt så är det en funktion som skall hålla kunderna uppdaterade om förändringar som kommer att ske med systemet och hur det påverkar användandet. Men det kan även tillhandahålla information om utbildningar och visningar av nya funktioner i systemet som användarna bör ha kunskap om för att kunna använda produkten på ett för den egna organisationen så effektivt sätt som möjligt.

#### **4.2.6.3 Telefonsupport**

Telefonsupport används framför allt för att ge stöd åt enskilda organisationer eller kunder när de har problem med att använda systemet. Det gäller då att kunna tillhandahålla specifik systeminformation för alla kunder som använder systemet, det finns nästan inte några kunder som använder systemet på samma sätt vilket komplicerar den här delen. Att då ha system som hanterar hur olika anpassningar är gjorda är en viktig faktor som framkommit vilket ger förtroende hos utvecklingsföretagets kunder.

#### **4.2.6.4 Ta emot krav, önskemål och felrapporter**

De problem som inte kan lösas via telefonsupporten hamnar istället under denna funktion. Kunderna kontaktar nästan uteslutande utvecklingsföretaget när de har problem med systemet och det är då viktigt att kunna hantera krav, önskemål och felrapporter på ett effektivt sätt. Det sker då initialt en prioritering av det som inkommit beroende på hur akut det är att åtgärda. Studiens respondenter framhöll felrapporter framför krav och till sist önskemål i sin prioritering av dessa funktioner.

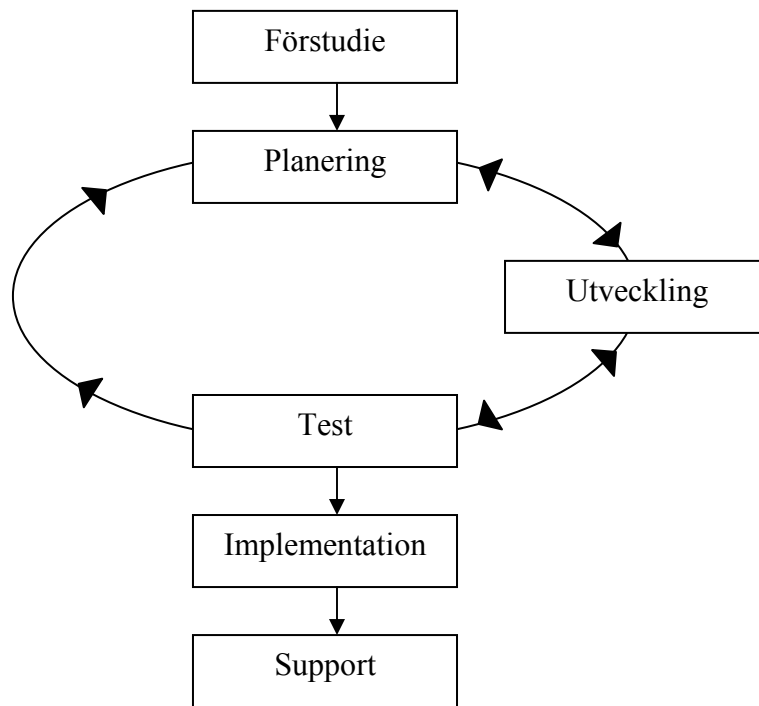
Användarföreningen har här en viktig roll och tillhandahåller redan filtrerad feedback på produkten som kan anses vara en mer allmän bild av hur kunderna ser på produkten. Användarföreningen ger även de kunder som är med ytterligare en bild av hur produkten kan användas i den egna verksamheten.

### **4.3 Utvecklingslivscykeln**

De delar som har presenterats i kapitel 4.2 hänger naturligtvis samman på något vis. Det utvecklingsflöde som respondenterna har beskrivit har här förts samman och presenteras i den uppfattade utvecklingscykeln i figur 4.9. Hur livscykeln för utveckling av standardsystem ser ut kan variera en del beroende på vilken fas standardsystemet befinner sig i produktlivscykeln (se figur 4.1). Olika delar i utvecklingslivscykeln blir mer eller mindre viktiga i olika faser av produktlivscykeln beroende på om det skapas en ny version av systemet, om det skapas en service pack eller en felkorrigering. Förstudie och Planering är mest framträdande i början av en produktlivscykel medan Implementation och Support mer blir en konsekvens av en färdig produkt.

Modellen kan på många sätt liknas vid en blandning av vattenfallsmodellen (se kapitel 3.1.1) och spiralmodellen (se kapitel 3.1.2) eftersom den består av delvis ett rakt flöde (förstudie – planering och test – implementation – support) och delvis ett iterativt flöde (planering – utveckling – test).

Resultat  
- *Konsten att göra det specifika generellt*



**Figur 4.9** Utvecklingslivscykeln

## 5 Diskussion och slutsats

Den undersökning som genomförts har varit i syfte att förstå hur utvecklingsprocessen för standardsystem ser ut, men även att förstå hur det kan skapas ett standardsystem som ska vara generellt utifrån de specifika användarsituationer som systemet ska användas i. De förkunskaper som funnits innan undersökningen påbörjades har varit fokuserade kring hur utvecklingen av konventionella system går till och hur de konventionella livscyklerna är strukturerade, vilket på ett flertal punkter har visat sig skilja sig markant från standardsystemsutvecklingen.

Vad som framkommit av undersökningen är att livscykeln för utveckling av standardsystem har både likheter och skillnader med de livscyklar vi undersökt som är utformade för konventionell systemutveckling (se kapitel 3.1). I figur 4.9 som är den livscykelmodell för standardsystemsutveckling som presenterats tidigare kan ses ett flertal element som överensstämmer mellan de olika modellerna.

De olika faser som återfinns i modellen för standardsystemsutveckling kan härledas bakåt till både vattenfallsmodellen och spiralmodellen. Dock är den inte fullt ut lik någon av de båda konventionella modellerna utan kan mer ses som en blandning av dem båda där processen börjar och avslutas med ett vattenfallsförlopp medan mitten av processen kan ses som en spiral.

Förstudiefasen i standardsystemslivscykeln har i princip samma funktion som förstudien i vattenfallsmodellen då det här byggs upp en informationsbas av den data som kommer behövas för projektet. Den här fasen är heller inget som påverkas av iteration utan något som enbart görs en gång, i början av projektet. Vad som också görs tydligt i denna fas är de skilda användarfokus som finns mellan de två utvecklingsätten, där konventionella system har lätt att samla in användarnas åsikter från en fördefinierad användargrupp, och där standardsystemsutvecklingen istället har en riklig mängd metoder för att samla in information från den marknad med diffust definierade användare som systemet befinner sig i. Denna information kommer också med ett annat perspektiv, som inte riktigt på samma vis kan härledas till den konventionella systemutvecklingen. Det marknadsmässiga perspektivet av att försöka veta vilken information som kan utveckla produkten vidare på något plan är och som på så vis är viktigt för produktens säljbarhet och påverkar då också företagets ekonomi både kort- och långsiktigt.

Iterationen som är en stor del av spiralmodellen kan även återfinnas i livscykelmodellen för standardsystem, skillnaderna är att inte hela processen itereras utan enbart faserna planering, utveckling och test. Att det ser ut på detta viset är på grund av att det i dessa faser finns en annan lösare koppling till de blivande användarna av systemet. Dessa faser sker till allra största del inom väggarna för utvecklingsföretaget och kan därför iterera fritt på samma sätt som iterationen sker inom spiralmodellen.

Diskussion och slutsats  
- *Konsten att göra det specifika generellt*

Att iterationen sedan upphör beror på att de saker som har passerat testerna skickas ut till företagets kunder för att implementeras på separata håll. Något som utvecklarna inte behöver vara inblandade i. I de flesta fall när det rör implementation av ett service pack eller en ny version så är detta något som görs av kunderna själva, och på så vis försvinner möjligheterna för att gå tillbaka till utvecklingsföretaget för att till exempel utföra nya tester. En annan intressant sak som vi har noterat med standardsystemsutvecklingen är att till skillnad från vattenfallsmodellen så sker testerna innan implementationen. Detta också givetvis kopplat till att det i ett standardsystem inte kan tillåtas att alla kunder installerar systemet för att det sedan ska kunna testas hos alla kunder, något som skulle bli alltför resurskrävande och opraktiskt för ett standardsystem, men som går att utföra med ett konventionellt system då det oftast endast handlar om att installera systemet på ett enda fall, eller i en organisation.

Efter implementationen följer en supportfas som verkar vara helt unik i sin uppbyggnad för standardsystemsutveckling, då en sådan varken finns med i vattenfallsmodellen eller spiralmodellen. Detta kan tänkas ingå i den konventionella systemutvecklingen sett som en service, vilket den gör i spiralmodellen, istället för en fas i livscykeln. Vi har dock valt att ha denna fas med i modellen för standardsystemsutveckling då det uppenbart är något som naturligt följer på implementeringsfasen av ett standardsystem i varje ny utvecklingscykel.

Förutom förstudiefasen så är det i testningsfasen som vi funnit stora skillnader för hur arbetet går till. I testningsfasen framgår det också tydligt att det är ett arbete som strävar efter att göra någonting generellt, men det framgår också här tydligt vilka problem det medför att utveckla standardsystem. I testningsfasen uppkommer nämligen tydligt problemen i att det är så gott som omöjligt att testa av alla fall som kan förekomma hos alla användare som finns av systemet. Detta skulle kräva specifikationer för alla kunders versioner av systemet, samt repliker av den miljö som de kör det i. Att sedan köra individuella tester av systemet i alla dessa testmiljöer är därför en orimlighet. Testningen får därför fokusera på att ta hänsyn till de vanligast förekommande problemen, samt att arbeta förebyggande och förlita sig på att systemet är gjort på så vis att det snabbt går att korrigera misstag som gjorts i utvecklingen.

Förstudiefasen och testningsfasen anses ha till största del svarat på ett av de förutsatta målen, hur det arbetas för att göra det specifika generellt, men även påvisat svårigheterna i just detta. En annan del av arbetet som skapade en viss förvåning var företagets brist på metoder för att arbeta mot att få ett generellt användargränssnitt. Att utveckla systemen på så vis att de blir generella ur en MDI-aspekt (Människa Dator Interaktion) saknas i de fall vi undersökt vilket vi fann underligt. Inget av företagen vi har varit i kontakt med har lagt ner tid och energi på denna delen av utvecklingen, mycket antagligen beroende på kunskapsbrist om vilka metoder som kan användas för att skapa sig en sådan bild av användandet som sker på systemet. Det närmsta ett steg i riktningen mot att försöka skapa ett standardsystem med MDI i baktanke som vi funnit har varit

utvecklandet av anpassningsbart användargränssnitt med hjälp av stylesheets där företagen har kunnat individualisera det grafiska gränssnittet av systemet.

Det vi funnit som viktigast i vår undersökning är den strävan som finns hos företagen vi undersökt efter att kunna samla in så mycket information som möjligt, genom användandet av ett flertal olika kanaler, för att i så stor omfattning som möjligt få veta vad användarna anser om det system som utvecklas. Detta givetvis drivet av rent marknadsmässiga intentioner, då systemet måste utvecklas efter kundernas önskemål för att kunna säljas. En drivkraft som borde bärga för att bättre system utvecklas, men också för att bättre metoder kommer att tas fram för att kunna utveckla dessa system på ett bättre sätt. Något som vi tror och hoppas att vi har kunnat givit ett bidrag till.

## **5.1 Reflektioner**

Syftet med denna uppsats har som tidigare diskuterats varit att utöka kunskapen om utvecklingsprocessen som ligger till grund för standardsystem. Då det inte har funnits så mycket material om ämnet så har uppsatsen kretsat mycket kring att försöka undersöka närliggande områden för att på så sätt försöka hitta tillräckligt med relevant information för att skapa en grund att bygga undersökningen runt.

Då problemen att finna information inom detta ämne har varit påtagliga har det visat sig mycket enklare att finna information om ämnen som ligger nära detta. Både livscykler, systemutveckling och standardsystem i allmänhet är omskrivna ämnen där det finns mycket information att inhämta. Här har det istället varit en fråga om att göra sorteringar av vad som är för uppsatsen relevant information.

De intervjuer som genomfördes befarades också vid ett tidigt stadium skapa en del bekymmer för uppsatsen, då ingen inblandad i uppsatsskrivandet har erfarenheter av liknande arbete. Detta ledde till omfattande förarbete, med en rad testinspelningar och testintervjuer för att förbättra våra färdigheter inom intervjukonsten. Att vi kommit väl förberedda, framförallt genom att ha tränat på att i intervjusituationer ställa för ämnet relevanta frågor, gjorde oss säkrare i intervjusituationen, samtidigt som den med största sannolikhet gjorde att vi snabbt kunde formulera, eller formulera om de frågor som vi ville ha svar på. Något som med största sannolikhet har höjt kvaliteten på undersökningen.

Vi tror oss ha uppfyllt de mål som vi strävat efter i att undersöka den process som leder fram till ett standardsystem på ett tillfredställande sätt, då det erhållits en stor mängd nya kunskaper inom detta ämne, kunskaper som vi på intet sätt har kunnat tillgodose oss genom idag befintlig litteratur.

## **5.2 Förslag till framtida forskning**

Under våra undersökningar så har vi stött på ett flertal olika saker som vi tidigare inte varit bekanta med, både inom och utanför det ämne som vi diskuterat.

Diskussion och slutsats  
- *Konsten att göra det specifika generellt*

En första sak som vi gärna skulle vilje se en fortsatt diskussion inom är olika former av standardsystemsutveckling och gärna då i form av djupare analyser av varje fas i utvecklingsprocessen. Då vår uppsats har valts att läggas på ett ganska så övergripande plan tror vi att det finns utrymme för att djupare gå in på varje enskild fas. De faser som vi finner vara extra intressanta är förstudie och testning som på många plan verkar skilja sig från de konventionella metoderna, och vara mer specifika för standardsystem.

I förstudiefasen finns en mängd olika varianter på hur information samlas in från en användargrupp som är svår att överblicka, en djupare analys av detta skulle vi anse vara intressant. I testningsfasen går systemen igenom en lång rad tester. Även här är det svårt att veta hur och vad som är viktigast att testa då det är omöjligt att täcka in alla fall som kan tänkas uppkomma i den mängd olika miljöer som kan finnas hos kunderna. En djupare analys för hur detta går till vore intressant.

Ett intressant begrepp som vi stött på under våra efterforskningar är user awards, som är en tävling för system och som vissa företag även anser vara en sorts testning. User awards är en amerikansk uppfinning som kommit till Sverige på senare tid. Detta är en tävling där ett användarföretag anmäler det system som de använder sig av till att delta. De system som sedan har anmälts testas och bedöms av en jury utifrån en rad olika aspekter. Systemen rankas sedan och belönas utifrån vilken position systemet slutade på. Detta tror vi skulle vara ett intressant ämne att undersöka utifrån perspektivet hur system bedöms och rankas. Ett annat projekt med bedömningar av standardsystem görs av ”Data Research för Data Produkt Utvärdering”. Mer information om detta går att finna på ”[www.dpu.se](http://www.dpu.se)”.

## **Bilaga 1: Intervjuguide**

För intervjuerna har vi tänkt behandla ett antal frågor som berör utvecklingsprocessen för standardsystem. Ordningen och formuleringen kommer att formas utefter hur intervjuerna framflyter. De personer som vi intervjuar har en ganska bred kunskap om företagets utvecklingsmodell. Frågorna som beskrivs här nedan är grunderna för intervjun och kommer att under intervjuns gång kompletteras med följdfrågor för att utröna underliggande kunskap och förtydliga svaret på den övergripande frågan.

### **Fråga 1**

Hur skulle du vilja beskriva företaget som du arbetar på?  
Din egen bakgrund, hur ser den ut?

Hur, Varför, När, Vem och Vad.

### **Fråga 2**

Det system som du är med och utvecklar, hur skulle du vilja beskriva det?

Hur, Varför, När, Vem, Vad

### **Fråga 3**

Har ni någon modell för utveckling av ert standardsystem och i så fall hur ser den ut?

Kan du beskriva utvecklingsmodellen?

Liknar den någon känd utvecklingsmodell?

Undersök dessa faktorer:

universality, confidence, experience, co-determination och introduction

Har ni testat andra utvecklingsmodeller?

Ge exempel?

Och varför är den ni använder bättre än andra eventuella?·Eller vad har fått er att använda den modell ni nu använder?

Hur, Varför, När, Vem, Vad

### **Fråga 4**

Vad är det som får er utvecklingsprocess att starta? (Eller starta om?)

Vad är det som gör att ni utvecklar ett nytt system, släpper en ny release eller uppdatering?

Andra faktorer är kundernas förfrågan?

Hur påverkar marknadsförändringar ert utvecklingsarbete?

Hur, Varför, När, Vem, Vad



**Fråga 5**

Själva utvecklingsprocessen, hur går den till?  
Varför gör ni så eller så?  
Eftersök en hög detaljnivå.

Hur, Varför, När, Vem, Vad

**Fråga 6**

Den design ni väljer är naturligtvis viktig för de som skall använda systemet, vilka faktorer avgör hur gränssnittet skall utformas?  
Hur ser affärslogiken ut i era system?

Hur, Varför, När, Vem, Vad

**Fråga 7**

När ni programmerar/kodar ert system, hur går ni då tillväga från kravspec till färdig programvara.

Hur, Varför, När, Vem, Vad

**Fråga 8**

Hur utvärderar ni er produkt?  
Finns det andra sätt som du tycker är lämpliga att använda?

Hur, Varför, När, Vem, Vad

**Fråga 9**

Gör ni någon riskanalys under utvecklingsprocessen för att ta kritiska beslut om projektets fortskridning?

Hur, Varför, När, Vem, Vad

**Fråga 10**

Bedriver ni era projekt i någon speciell organisationsform?  
Hur ser den i så fall ut?  
Eller  
Nej. Hur går in då tillväga?

Hur, Varför, När, Vem, Vad

**Fråga 11**

Hur involverar ni användarna i utvecklingsprocessen?

Hur, Varför, När, Vem, Vad

Bilagor  
- *Konsten att göra det specifika generellt*

***Vilka personer frågar vi:***

Projektledare  
Utvecklare  
Programmerare  
Gränssnittsdesigner  
Marknadschef

Om systemet, vilken typ av system o.s.v. som utvecklas?  
Organisation?

***Intressanta områden:***

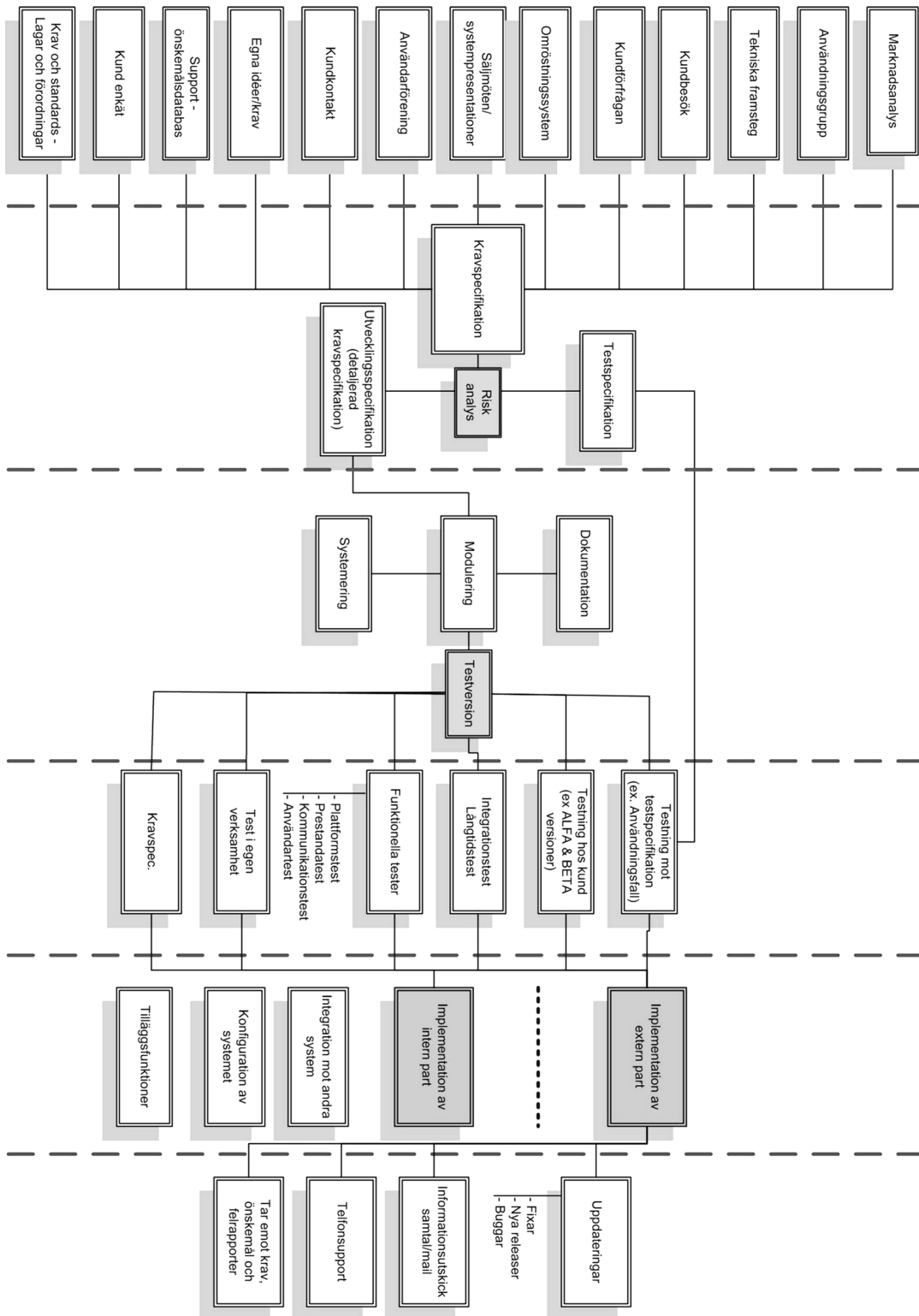
Kravspeg.  
Utveckling  
Design  
Implementation/kodning  
Test  
Riskanalys – när och hur vet man om man ska fortsätta med ett projekt/del  
av system?

Metoder

Användare – nuvarande och framtida

Bilagor  
- *Konsten att göra det specifika generellt*

**Bilaga 2: Causal network**



## 6 Källförteckning

- Alter, S. (1999) *Information Systems: A Management Perspective*, Addison-Wesley, US.
- Andersson, B-E. 2001. *Som man frågar får man svar*. Prisma, Stockholm
- Andersen, N.-E., Kensig, F., Lassen, M., Lundin, J., Mathiassen, L., Munk-Madsen, A. & Sødergaard, P. (1998) *Professional Systemudvikling*. Teknisk Forlag A/S, København.
- Bryman, A. 1997. *Kvantitet och kvalitet i Samhällsvetenskaplig forskning*, Studentlitteratur, Lund.
- Cugola, G., Ghezzi, C. (1998). *Software processes: A retrospective and a part to the future. Software process-improvement and practice*, vol 4. 101-123.
- Carmel, E. (1993). *A discussion of special characteristics for software package development life cycle models*. *Software engineering notes*, vol. 18 no 2. 23-24
- Carmel, E. & Sawyer, S. (1998). *Packaged software development teams: what makes them different?* *Information technology & People*, vol. 11. pp. 7-19, MCB University Press.
- Cooper, R., Zmud, R. (1990). *Information technology Implementation Research: A Technological Diffusion Approach*. *Management Science*, 36 (2), 123-139.
- Checkland, P. & Holwell, S. (1998). *Information, Systems and Information Systems*. West Sussex, England.
- Denscombe, M. (1998). *Forskningshandboken – för småskaliga forskningsprojekt inom samhällsvetenskaperna*. Studentlitteratur, Lund.
- Eriksson, L T, Wiedersheim-Paul, F. 1997. *Att utreda, forska och rapportera*. Liber Ekonomi, Malmö.
- George, J. F. (2000). *The origins of software: Acquiring Systems at the End of the Century. Framing the domains of IT Management: Projecting the Future... Through the Past*. Cincinnati, Ohio: Pinnaflex educational resources.
- Gilovich, T. (1991). *How we know what isn't so*, The Free Press
- Hansen, B., Kautz, K. & Jacobsen, D. (2003) *Information System Development Methodologies in Practice. International Conference on Information System Development 2003, Melbourne*.
- Iivari, J., Hirschheim, R. & Klein, H. (1998) *A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. Information Systems Research*. Vol. 9, No. 2, June 1998.
- Klaus, H., Rosemann, M., & Gable, G. C. (2000). *What is ERP? Information System Frontiers*, 2000.
- Kruchten, P. (2000). *From waterfall to iterative lifecycle – a tough transition for project managers*. Rational Software Corporation.
- Kvale, S (1997), *Den kvalitativa forskningsintervjun*, Studentlitteratur, Lund

Källförteckningr  
- *Konsten att göra det specifika generellt*

- Kwon, T. K. and R. W. Zmud (1987). Unifying the Fragmented Models of Information Systems Implementation. *In Critical Issues in Information Systems Research* (R.J. Boland and R.A. Hirschheim Eds.), pp. 227-251, John Wiley and Sons, New York.
- Markus, M. L. (2000). Paradigm Shifts – E-Buisness and Buisness/Systems Integration. *Communications of the AIS*.
- Miles, M B, Huberman, A M. 1994. *Qualitative Data Analysis*, Sage Publications
- Nilsson, A. G. (1991). Anskaffning av Standardsystem för att utveckla verksamheter: Utveckling och prövning av SIV-metoden. Stockholm.
- Patel, R. Davidsson, B. (1991). *Forskningsmetodikens grunder – Att planera, genomföra och rapportera en undersökning*. Studentlitteratur, Lund.
- Sawyer, S. (2000). Packaged software: implications of the differences from custom approaches to software development. *European Journal of Information Systems*, Vol. 9, 47-58.
- Scott J. & Vessey, I (2000). Implementing Enterprise Resource Planning Systems: The Role of Learning from Failure, *Information Systems Frontiers*.
- Vamanana, M., Wangb, Q., Battab, R., Szczerbad R. J. (2002) Integration of COTS software products ARENA & CPLEX for an inventory/logistics problem.
- Wallén, G. (1993). Vetenskapsteori och forskningsmetodik, Studentlitteratur
- Yin, R K. (2003), Case Study Research, Design and Methods, Sage Publications