

Nackdelar med sekventiella utvecklingsmodeller

Resultatet av omedvetna val

Magisteruppsats, 10 poäng, inom Systemvetenskapliga programmet

Framlagd: Januari, 2005

Författare: Niklas Bendelius, 790330-0015
Mattias Jönsson, 790108-3936

Handledare: Odd T. Steen

Nackdelar med sekventiella utvecklingsmodeller

Resultatet av omedvetna val

© Niklas Bendelius, Mattias Jönsson

Magisteruppsats framlagd: januari 2005

Omfång: 90

Handledare: Odd Steen

Resumé: Författarna har upplevt att systemutvecklingspraxis ofta baseras på sekventiella systemutvecklingsmodeller. Med undersökningen vill författarna visa på vilka nackdelar detta eventuellt för med sig för projektets resultat. Med resultat avses två saker; dels hur nöjd kunden är med projektets utfall och dels hur väl projektet har hållits inom givna resursramar. Undersökningen är genomförd som en fallstudie med tre fall, baserad på en genomgång av tidigare forskning inom området.

Genomgången av tidigare forskning resulterade i fyra grupper av nackdelar varav tre kunde verifieras av fallstudien. Förutom att kunna verifiera problem att möta förändring i krav såväl som i förutsättningar för arbetet och användardistansering fann författarna en ökad risk för negativa konsekvenser av brister i medarbetarnas kompetens och en risk att externa personers insyn i projektet försämras.

Slutligen resonerar författarna kring det problematiska i att komma till rätta med nackdelarna relaterat till de sekventiella systemutvecklingsmodellerna då dessa ofta är omedvetna och snarast ses som ett naturligt sätt att lägga upp arbetet än som ett medvetet val.

Nyckelord: Systemutvecklingsmodell, Sekventiellt arbetssätt, Omedvetna val

Innehållsförteckning

1. INLEDNING	1
1.1. VAD ÄR ETT SYSTEMUTVECKLINGSPROJEKT?	1
1.2. NACKDELAR? FÖR VEM?	2
1.3. MODELL KONTRA PROCESS	3
1.4. FRÅGESTÄLLNING	4
1.5. SYFTE	5
1.6. AVGRÄNSNINGAR	5
1.7. DISPOSITION	6
2. TILLVÄGAGÅNGSSÄTT	7
2.1. FOKUS SAMT INLEDANDE DEFINITIONER	7
2.1.1. KONCEPTUELLT RAMVERK	7
2.1.2. UNDERSÖKNINGSFRÅGOR	9
2.1.3. UNDERSÖKNINGSSTRATEGI	10
2.1.4. URVAL	11
2.2. DATAINSAMLING	12
2.2.1. INSTRUMENTATION	13
2.2.2. DATAINSAMLINGSMETODER	13
2.3. ANALYS AV DATA	15
2.4. ETIK OCH KVALITET I UNDERSÖKNINGEN	16
2.4.1. ETISKT RAMVERK	16
2.4.2. VALIDITET	17
2.4.3. RELIABILITET	17
2.5. RAPPORTEN	17
3. SYSTEMUTVECKLINGSMODELLER OCH -PROCESSER	19
3.1. SEKVENTIELLA MODELLER	19
3.1.1. EXEMPEL PÅ SEKVENTIELLA SYSTEMUTVECKLINGSPROCESSER	22
3.2. EVOLUTIONÄRA MODELLER	22
3.2.1. SPIRALMODELLEN	23
3.2.2. ITERATIV UTVECKLING	23
3.2.3. EXEMPEL PÅ EVOLUTIONÄRA SYSTEMUTVECKLINGSPROCESSER	24
3.3. NACKDELAR MED DE SEKVENTIELLA MODELLERNA	25
3.3.1. FÖRÄNDRINGAR I SYSTEMKRAVEN	25
3.3.2. FÖRÄNDRINGAR I FÖRUTSÄTTNINGAR	26

3.3.3.	FOKUS PÅ DOKUMENTATION	27
3.3.4.	TEKNISKT FOKUS OCH ANVÄNDARDISTANSERING	28
3.4.	FINNS SAMMA NACKDELAR HOS DE EVOLUTIONÄRA MODELLERNA?	29
3.5.	SAMMANFATTNING	30
<u>4. FALLSTUDIEN</u>		<u>31</u>
4.1.	STRUKTUR	31
4.2.	FALL ETT	31
4.2.1.	FALL ETT – BESKRIVNING	31
4.2.2.	FALL ETT - DATAINSAMLING	32
4.2.3.	FALL ETT - UTFALL	32
4.3.	FALL TVÅ	35
4.3.1.	FALL TVÅ – BESKRIVNING	35
4.3.2.	FALL TVÅ – DATAINSAMLING	35
4.3.3.	FALL TVÅ – UTFALL	36
4.4.	FALL TRE	38
4.4.1.	FALL TRE - BESKRIVNING	38
4.4.2.	FALL TRE - DATAINSAMLING	38
4.4.3.	FALL TRE - UTFALL	38
4.5.	SAMMANSTÄLLNING	40
<u>5. ANALYS OCH REFLEKTION</u>		<u>43</u>
5.1.	FÖRÄNDRINGAR I SYSTEMKRAVEN	43
5.2.	FÖRÄNDRINGAR I FÖRUTSÄTTNINGARNA	44
5.3.	FOKUS PÅ DOKUMENTATION	46
5.4.	TEKNISKT FOKUS OCH ANVÄNDARDISTANSERING	47
5.5.	NACKDELARNAS PÅVERKAN PÅ RESULTATET	47
5.6.	FINNS DET FLER NACKDELAR?	50
<u>6. AVSLUTANDE DISKUSSION OCH PRESENTATION AV RESULTAT</u>		<u>52</u>
6.1.	ATT KNYTA IHOP SÄCKEN	52
6.1.1.	GRAD AV PÅVERKAN PÅ RESULTATET	52
6.1.2.	VARFÖR VÄLJER MAN DÅ EN SEKVENTIELL MODELL?	53
6.2.	SLUTSATSER	54
6.3.	REFLEKTION ÖVER ARBETET	55
6.4.	VIDARE FORSKNING	55
<u>7. BILAGOR</u>		<u>56</u>
<u>8. KÄLLFÖRTECKNING</u>		<u>82</u>
8.1.	LITTERATUR	82
8.2.	ARTIKLAR	82
8.3.	ÖVRIGT MATERIAL	83

1. Inledning

Systemutveckling idag grundar sig enligt vår mening ofta på sekventiella tankemodeller, eller systemutvecklingsmodeller, vilka kan föra med sig nackdelar för projektet. Begreppet sekventiella systemutvecklingsmodeller kommer vi att behandla mer ingående senare i rapporten, men för att läsaren lättare ska kunna följa med i de första kapitlen vill vi förtydliga vad vi menar med detta. En systemutvecklingsmodell är alltså den generella ansats man har för att lösa ett visst problem, alltså på en mer generell nivå än en enskild systemutvecklingsmetodologi/systemutvecklingsprocess. En sekventiell systemutvecklingsmodell är således vidare en modell av detta slag som ser systemutvecklandet som ett arbetssätt där man steg för steg arbetar sig fram till ett slutresultat. Dessa steg, eller faser, låter arbetet i varje enskild fas bygga på arbetet och resultatet från föregående fas. Nedan ämnar vi i vår undersökning inledningsvis försöka förklara vad vi menar med detta, underbygga vårt resonemang samt redogöra för en del av de begrepp som vi använder oss av vidare i uppsatsen.

Det påstås vi gör ovan att sekventiella systemutvecklingsmodeller som bas för ett systemutvecklingsprojekt skulle medföra nackdelar och potentiella problem understöds av flera författare, bland annat Beck (2001), Kruchten (2002), Boehm (1988) och Cotton (1996). Dessa författare menar bland annat följande: ”*Den sekventiella processen, eller vattenfallsprocessen, fungerar inte.*” (Kruchten, 2002, s. 57) Vidare hävdar Boehm (1988) att ”[The waterfall model] *does not work well for many classes of software, particularly interactive end-user applications.*” Trots detta så menar Gulliksen & Göransson (2002) att vattenfallsmodellen idag är den vanligaste systemutvecklingsmodellen att använda som grund för systemutvecklingsprojekt.

Detta ligger i linje med vad vi som författare har upplevt då vi kommit i kontakt med ett antal systemutvecklingsprojekt. Vi upplever dock att dessa nackdelar väldigt sällan tas upp för diskussion eller reflektion i det praktiska planerings- och utvecklingsarbetet. Genom att visa på vilka nackdelar som finns med de sekventiella modellerna vill vi öka medvetenheten hos morgondagens systemvetare kring modeller och deras nackdelar.

1.1. Vad är ett systemutvecklingsprojekt?

Inledningsvis vill vi definiera vad vi menar med ett systemutvecklingsprojekt. Begreppet är uppbyggt av två delar: systemutveckling och projekt, där vart och ett av dessa begrepp måste definieras för att man ska få en definition av helhetsbegreppet. Andersen et al. (1994) menar att begreppet projekt syftar på en arbetsform som för det första är en engångsuppgift. Detta innebär att varje projekt är unikt och även om man i slutändan siktar på att nå samma resultat i två olika projekt så är vägen dit alltid olika. Vidare menar författarna att man ska ha ett på förhand uppställt mål. Detta mål är i sig projektets existensberättigande vilket innebär att om inte målet funnits hade inte heller projektet startats överhuvudtaget. Avslutningsvis menar de att ett projekt alltid är tidsbegränsat.

Andra karakteristika som återfinns i litteratur är till exempel det faktum att ett projekt alltid har en beställare eller kund. Denna kan både finnas internt i den egna organisationen eller externt, till exempel en betalande kund. Denna kund ställer tveklöst krav på projektet och projektdeltagarna antingen genom de resurser de kan tänka sig att investera i projektet eller genom de förväntningar de har på projektets resultat. (Gulliksen & Göransson, 2002)

Gulliksen & Göransson (2002) menar att begreppet systemutveckling inbegriper flera olika delar; till exempel utveckling av mjukvara, hårdvara, dokumentation och arbetsorganisation, vilket också stöds av andra författare som till exempel Alter (1996). Alter menar till exempel att man måste göra en distinktion mellan informationsteknologi och informationssystem där informationssystemet innefattar allt från organisationen till varors streckkoder medan informationsteknologin är den hård- och mjukvara som möjliggör ett utnyttjande av informationssystemet. Systemutveckling är alltså alla aktiviteter som syftar till att effektivisera eller förbättra någon form av informationssystem. Detta leder i sin tur till att ett systemutvecklingsprojekt är ett projekt i vilket man utför ovan nämnda utveckling, ofta med hjälp av informationsteknologi.

1.2. Nackdelar? För vem?

När man påstår att någonting medför nackdelar är det viktigt att man inser problematiken med att påstå detta. Vi menar att sekventiella systemutvecklingsmodeller medför nackdelar relaterat till projektets resultat och avgränsar oss i vår undersökning till dessa nackdelar. Men hur kan man avgöra huruvida man uppnått ett bra resultat? Om projektet till exempel leder till att den beställande avdelningen i organisationen når ett sämre resultat kanske detta kan ses som ett bra resultat för konkurrerande avdelningar? Kanske är en nackdel för en del av organisationen en fördel för en annan? Det är alltså inte alltid så lätt att säga om projektet är lyckat eller inte, eller för den delen om en nackdel verkligen är en nackdel.

Andersen et al (1994) menar bland annat att det är viktigt att projektet ska ligga i linje med organisationens övriga verksamhet och att man på förhand har satt upp tydliga mål med projektet. Vi ser en poäng i detta eftersom man faktiskt kan säga att ett projekt var lyckat om det mötte målet eller målen med projektet.

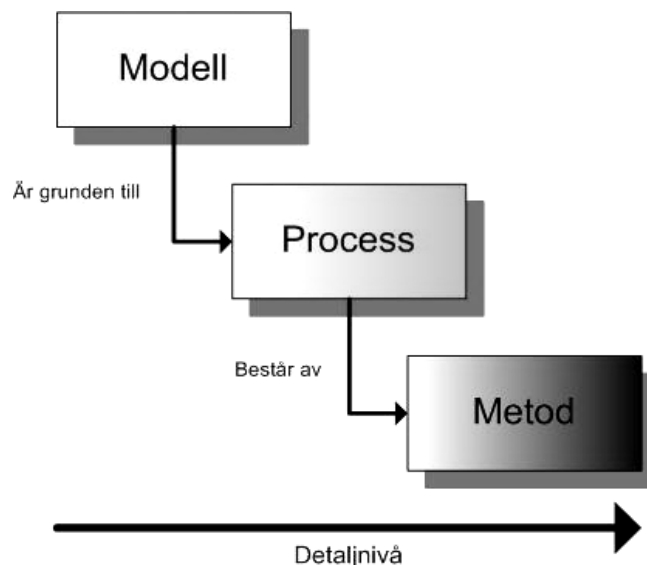
Vi måste dock lägga på ytterligare kriterier, däribland att man ska ha uppnått målet på utsatt tid och med de resurser som avsattes, annars kan man inte säga att projektet var lyckat till hundra procent. Dvir & Lechler (2004) menar dock att denna definition på huruvida ett projekt har lyckats eller ej både är kraftigt förenklad och vilseledande. Man saknar här aspekter som till exempel hur nöjd kunden är med det levererade resultatet eller hur nöjda användarna är med systemets användargränssnitt. Dvir & Lechler presenterar istället ett antal forskares teorier om hur man bör mäta hur ett projekt lyckats. Shenhar et al. (1997 i Dvir & Lechler, 2004) har till exempel identifierat tre aspekter: "Meeting design goals", "Benefits to customers" och "Commercial success and future potential", vilka vi anser täcka in en del av aspekterna men inte alla. Vi kan till exempel här inte se huruvida man hållit budget eller dragit ut på tiden. Istället kan vi tycka att Lipovetsky et al.'s (1997 i Dvir & Lechler, 2004) fyra dimensioner är betydligt klarare: "Customer satisfaction", "Efficiency" (som bland annat innebär ett mått för hur väl man hållit tidsplan och budget), "Commercial success" och "Future potential". Författarna menar dock att de två sista dimensionerna i princip är helt obetydliga, och vidare

att den första, "Customer satisfaction" var mer än dubbelt så viktig som "Efficiency". Därför är nackdelar som påverkar de kriterier vi nämner ovan, framför allt hur nöjd kunden är med resultatet, betydande för utgången av ett systemutvecklingsprojekt.

Då man ställer sig frågan hur man bör göra för att uppnå så hög chans som möjligt för att uppnå framför allt en hög grad av "Customer satisfaction" kommer vi in på så kallade systemutvecklingsprocesser. I dag finns ett antal olika sådana processer som var och en hävdar att just den är den bästa processen att använda för att hålla hög kvalitet och undvika de fallgropar som vissa forskare menar kan föreligga i ett systemutvecklingsprojekt. Bland dessa återfinns till exempel Rational Unified Process (RUP), Extreme Programming (XP) och Strukturerad Analys och Konstruktion av informationsbehandlingssystem (SAK), varav alla har sin grund i någon form av konceptuell utvecklingsmodell för hur systemutvecklingsarbetet ser ut från början till slut.

1.3. Modell kontra process

Vad är då egentligen denna konceptuella systemutvecklingsmodell kontra en systemutvecklingsprocess? Är båda inte egentligen bara andra namn på begreppet metod? Hull et al. (2002. Se även Gulliksen & Göransson, 2002) menar här att begreppen modell, metod och process ofta används mycket slarvigt och inte alltid helt korrekt. Begreppet modell är ett generellt beskrivande av hur verkligheten ser ut, medan en process beskriver ett antal strukturerade händelser som tillsammans ska åstadkomma ett specifikt mål. Så medan en modell är deskriptiv är en process mer normativ och försöker förmedla *hur* man bör göra. Vidare kan skillnaden mellan process och metod vid första anblicken verka ganska subtil. En metod är, till skillnad från processen som beskriver ett antal händelser, en specifik beskrivning av ett tillvägagångssätt för att utföra en aktivitet. En systemutvecklingsprocess skulle således kunna involvera ett antal metoder för att utföra en aktivitet. (Hull et al, 2002) För att ytterligare tydliggöra vad vi menar med begreppen, och hur de är relaterade till varandra visar vi en enkel bild i figur 1.1.



Figur 1.1 – Skillnad på modell, process och metod (egen konstruktion)

En modell är alltså en grundläggande ansats till hur ett problem löses; ett sätt att förhålla sig till problemet, medan processen ligger på en mer detaljerad nivå. Det är också här som denna uppsats primära fokus ligger; vi tittar på olika ansatser snarare än olika systemutvecklingsprocesser. Dessa processer, där man ser exempel som RUP och Extreme Programming, kallas också av vissa författare ”metodologier”, vilket vi dock anser vara en aning vilseledande. Russo & Stolterman (2000) talar om just metodologier och menar att användandet av dessa i grund och botten är omtvistat. Författarna menar att många organisationer hävdar att de använder någon typ av process, men att de flesta av dessa organisationer faktiskt inte använder dessa processer fullt ut i praktiken; processerna förändras lite ”on-the-go” för att passa den aktuella situationen (Russo & Stolterman, 2000). Detta innebär att en studie av de underliggande systemutvecklingsmodellerna är betydligt intressantare att titta på än praktiska processer, då systemutvecklingsmodellerna också ligger till grund för de modifieringar man gör i processerna.

Metoder är avslutningsvis särskilda tekniker som används för ett visst, mycket specifikt syfte, som ofta existerar som delar i en större process. Några exempel på metoder kan vara UML-modellering eller Storyboarding.

Tittar man på hur systemutvecklingsmodellerna har utvecklats med tiden ser man ganska stora skillnader. Från att ha gått från ett mycket enkelt, och kanske framför allt tekniskt fokuserat, angreppssätt med själva programmerandet som det centrala i systemutvecklingsprojektet där man egentligen löser problem genom att helt enkelt skriva lite mer kod, till väldigt dynamiska modeller med olika delar som itereras igenom för att på så sätt framskrida i sitt projektarbete. De klassiska motpolerna är den modell man brukar kalla vattenfallsmodellen och de modeller som har mer iterativt fokus, till exempel den evolutionära utvecklingsmodellen. Vattenfallsmodellen är i grund och botten en väldigt enkel sekventiell modell där varje steg, eller fas, utförs efter varandra. I ursprungsmodellen har man inte möjlighet att gå tillbaka och göra några ändringar i de färdigställda produkter som varit resultatet av tidigare faser, vilket potentiellt kan leda till att man är mycket noggrann i sitt arbete med en viss fas. Detta är dock den största kritik som också riktats mot vattenfallsmodellen, men detta kommer vi att gå in mer på senare. (Gulliksen & Göransson, 2002).

De iterativa, eller evolutionära, modellerna har inga faser på samma sätt som de sekventiella modellerna. Man har här inte projektdelar som utförs sekventiellt utan många delar av projektet flyter in i varandra och utförs ofta till och med helt parallellt. Detta kan tyckas rörligt och kan verka leda till ett planlöst irrande för att uppnå projektets mål, vilket dock många forskare inte skulle hålla med om. Fokus hos forskare idag verkar istället framför allt ligga på dessa senare typer av modeller där man har ett mer iterativt och parallellt arbetssätt. Detta tar sig uttryck i processer som RUP där man bland annat arbetar med analys, design och konstruktion parallellt (Kruchten, 2002) och XP där man genom sina många korta iterationer uppnår ett mer parallellt flöde av aktiviteter (Beck, 2001).

1.4. Frågeställning

Trots att det alltså finns forskare och andra aktörer som visar på behovet av ett mer parallellt arbetssätt så bygger många systemutvecklingsprojekt idag på en sekventiell modell som grund (Apelkrans & Åbom, 2001). Gulliksen & Göransson (2002) menar till och med att processer som bygger på en

sekventiell modell, som till exempel SAK, är de vanligaste i svenska systemutvecklingsprojekt idag. I och med detta blir det intressant att ställa sig frågan vad det kan finnas för negativa aspekter med detta så att man som utvecklare i ett projekt som bygger på en sekventiell modell kan ha det med sig i sitt arbete för att försöka minimera nackdelarnas negativa inflytande på resultatet.

Vår frågeställning lyder alltså:

Vilka nackdelar finns, relaterat till resultatet, med att grunda arbetet i ett systemutvecklingsprojekt på en sekventiell systemutvecklingsmodell?

1.5. Syfte

Syftet med denna undersökning är dels att göra en inledande litteraturundersökning i vilken vi sammanställer vad olika forskare och författare skriver om att ha en sekventiell modell som grund för arbetet i ett systemutvecklingsprojekt och dels att underbygga den undersökningen med en fallstudie i vilken vi undersöker ett flertal sådana projekt, i vilka man använt sig av ett sekventiellt arbetssätt, för att försöka visa på de eventuella negativa aspekter som detta förde med sig. Detta för att leda oss fram till våra slutsatser som svar på vår frågeställning ovan.

1.6. Avgränsningar

Den största avgränsningen vi har gjort är relaterat till resultatet. Att analysera något i förhållande till projektets resultat kan vara mycket komplext, då det kan vara mycket svårt att se alla aspekter i ett så pass intrikat begrepp som ”resultat”. Vi har således valt att göra vår undersökning relaterat till de aspekter av resultatet som vi talat om i kapitel 1.2, och inte alls fokusera på andra eventuella aspekter av resultatet.

Vi har dessutom valt att inte heller genomföra intervjuer med kunderna eller att på något annat sätt få förstahandsinformation kring dessas uppfattning. Detta trots att vi utgår från att det mest centrala i projektets resultat är ”Customer Satisfaction”. Det finns två huvudskaliga motiv till detta: För det första har vi i vissa fall (till exempel fall tre) haft mycket svårt att få möjlighet att genomföra några intervjuer med kundföretaget och för det andra kände vi att projektdeltagarna i många av fallen i stor utsträckning hade en god uppfattning om kundernas åsikter.

Vidare kan man eventuellt även se de val som legat till grund för vår frågeställning som avgränsningar i viss mån. Vi har, som synes i frågeställningen, endast valt att titta på nackdelar med sekventiella modeller och inte med några andra typer av modeller. Vi har också valt att enbart undersöka nackdelar med systemutvecklingsmodeller och inte med enskilda systemutvecklingsprocesser. Detta resonemang grundar sig till viss del i det som Russo & Stolterman (2000) menar i sin artikel. Eftersom systemutvecklare uppenbarligen inte alltid följer en systemutvecklingsprocess fullt ut är det betydligt intressantare att titta på de ansatser som ligger till grund för dessa processer.

Slutligen har vi valt att titta på traditionella systemutvecklingsprojekt där man börjar från grunden och utvecklar ett system som svar på specifika behov hos kund. Vi kommer inte att låta införandeprojekt av ERP-system eller andra specialfall ingå i vår undersökning. Detta för att vi ska kunna fokusera på modellens inverkan på arbetet och slippa ta hänsyn till den aktuella uppgiftens påverkan.

1.7. Disposition

Vi har valt att strukturera rapporten med en vanlig linjär disposition, där vi inleder med att bland annat redogöra för bakgrunden till och syftet med undersökningen för att sedan gå vidare och i kapitel två beskriva den metod vi använt oss av. Därefter diskuterar vi tidigare forskning på området. Denna teori ligger därefter till grund för den empiriska undersökning vi utfört vars resultat presenteras och analyseras i nästföljande kapitel. Avslutningsvis diskuterar vi de resultat vi kommit fram till och lägger fram våra slutsatser och även förslag på framtida forskning inom samma område.

2. Tillvägagångssätt

Nedan presenteras vår metod som har väglett vårt undersökningsarbete och som legitimerar våra resultat. Vi har valt att lägga upp det på så sätt att vi inleder med att presentera den strategi som vi har använt oss av i arbetet följt av det fokus vårt arbete har. Vidare så presenterar vi metoder för datainsamling och analys för att slutligen förklara våra tankar kring själva rapporten (uppsatsen) och vår syn på etik i vårt arbete.

2.1. Fokus samt inledande definitioner

I detta underkapitel ska vi försöka visa på det fokus vi haft under vårt arbete och göra lite inledande definitioner.

2.1.1. *Konceptuellt ramverk*

I undersökningen användes ett konceptuellt ramverk för att förklara vilka objekt som skulle studeras, nyckelfaktorer och variabler samt de relationer som man kan uppfatta mellan dessa. Detta verktyg användes för att vi skulle få en klarare bild av vad det var vi skulle undersöka och vilka olika delar som fanns med i vår undersökning. Det konceptuella ramverket mer eller mindre tvingade oss att vara selektiva, att välja mellan vilka variabler i undersökningen som var viktigast, vilka relationer som var mest intressanta och som en konsekvens av detta även vilken typ av data som skulle samlas in och analyseras. (Miles & Huberman, 1994)

Inledningsvis skapade vi oss en översiktlig bild av hela studien med vilken vi ämnade visa hur de olika fallen i vår fallstudie hörde ihop och hur de tillsammans kom att leda oss fram till våra slutsatser. Arbetet kom att ske i vart och ett av fallen med att samla in och analysera data. De data som samlats in och analyserats användes sedan i en fallövergripande analys som ledde oss fram till vårt undersökningsresultat. Denna översiktliga bild är den övre delen av vårt konceptuella ramverk. Vi har, som synes i modellen nedan, använt oss av tre ”units of analysis” (Yin, 2003), som valts ut för att matcha våra kriterier på ett fall (till exempel att de ska ha använt sig av en sekventiell utvecklingsmodell som grund för arbetet). Dessa kriterier redogörs för nedan.

De två primära entiteterna i varje fall är utvecklingsmodellen och nackdelarna som denna för med sig, eftersom det är detta vår undersökning kretsar kring. Vidare kommer vi att titta på projektarbetet eftersom det är via den aktiviteten som vi kommer att kunna dra fram de eventuella nackdelarna som systemutvecklingsmodellen för med sig. Slutligen kommer vi även att titta på delar av resultatet, ”Customer Satisfaction” och ”Efficiency”, för att visa på att det som vi pekar på som nackdelar verkligen är just nackdelar. Med Customer Satisfaction menar vi flera saker, bland annat är det ett mått för hur nöjd kunden är med resultatet med hänsyn till användbarhet och till hur väl det uppfyller

ställda krav och mål. Även med Efficiency menar vi flera saker men de kan summeras som hur väl projektet uppfyllde sina mål med givna resurser, bland annat i form av till exempel tidsplan och budget. Anledningen att vi valt just dessa två delar av resultatet är att det är de som är mest betydelsefulla. (Lipovetsky et al, 1997)

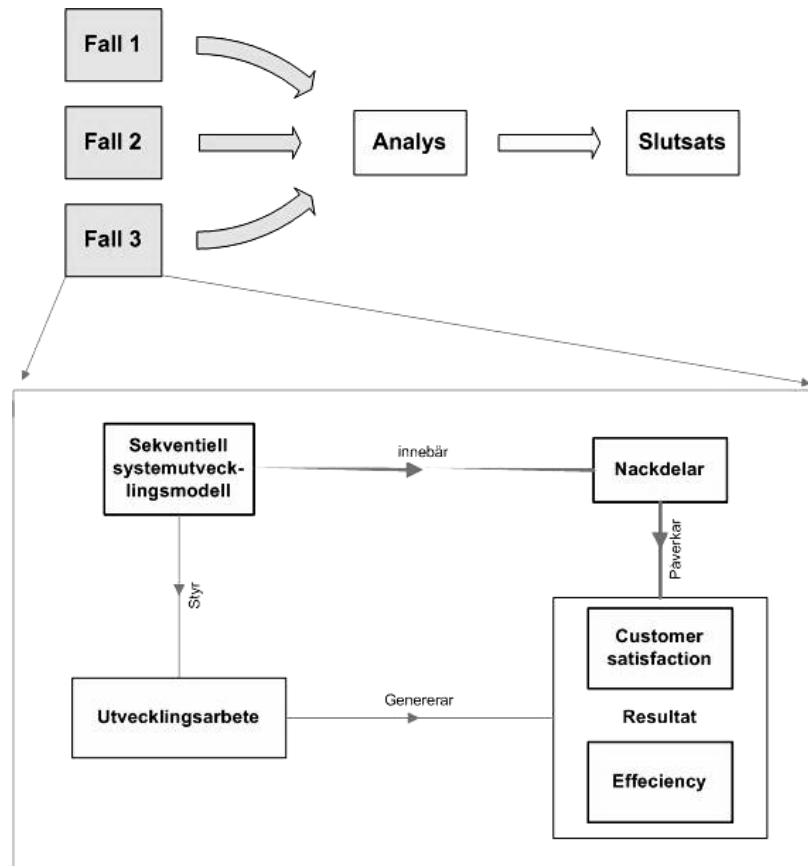
När vi nu specificerat våra entiteter är det dags att visa vilka relationer dem emellan som är intressanta. För det första så resulterar den grundläggande systemutvecklingsmodellen i ett antal nackdelar. Detta är centralt för vår studie eftersom vi har undersökt vilka nackdelar en sekventiell modell kan ge upphov till. Dessa nackdelars påverkan på resultatet är också centralt eftersom det är detta som visar att det är just nackdelar vi hittat. I själva verket är det nackdelarnas påverkan på arbetet som ger en påverkan på resultatet men för att tydliggöra vad det är vi syftar på har vi valt att göra just den relationen specifik i modellen och därför ritat in den mellan nackdelar och resultat.

Vidare så styr ju systemutvecklingsmodellen utvecklingsarbetet eftersom det är modellen som ligger till grund för hela arbetet. Detta är inte centralt för studien men finns med för att visa på hur entiteterna hör samman. Detsamma gäller för relationen som visar på det faktum att utvecklingsarbetet genererar ett resultat.

Vi anser att det är av största vikt att vi fokuserar på det som är centralt i vår undersökning, alltså modellen och nackdelarna. Att specificera alla andra möjliga och omöjliga variabler i kontexten som också kan ha påverkat resultatet, och på så sätt skulle kunna göra det svårare för oss att se nackdelarnas direkta påverkan, skulle bara leda till att ytterligare dimensioner infördes i vårt arbete som inte direkt är relaterade till undersökningens syfte. Detta anser vi skulle riskera att påverka undersökningens resultat negativt. Istället kommer vi att vara noggranna i vår undersökning för att kunna vara säkra på att relationen mellan nackdelarna och resultatet är verklig.

Vi var alltså inte intresserade av kontexten, såsom till exempel uppgiften. Vi var inte heller intresserade av att dela upp utvecklingsarbetet i de faser som det har bestått av eftersom det är helheten av modellens påverkan som är av intresse. Med andra ord har undersökningen begränsats till de entiteter och relationer vi tar upp i vårt konceptuella ramverk i varje enskilt fall. Inga andra entiteter, som till exempel endast indirekt påverkar resultatet, har inkluderats i undersökningen.

Vårt konceptuella ramverk visas i figur 2.1 nedan:



Figur 2.1 – Konceptuellt ramverk

Självklart så har detta ramverk reviderats vartefter undersökningen fortskred och vi fick en klarare bild över hur projektorganisationerna vi undersökte såg ut, och vilka entiteter och relationer som verkligen fanns. (Miles och Huberman, 1994)

2.1.2. Undersökningsfrågor

Resonemanget ovan i arbetet med det konceptuella ramverket ledde fram till ett antal undersökningsfrågor. Undersökningsfrågor använde vi i vår studie för att göra våra teoretiska antaganden mer explicita och för att komma fram till vad det var vi ville undersöka först. De data vi samlade in blev också troligen mer fokuserade och begränsade i sin omfattning. (Miles & Huberman, 1994)

Vi valde att arbeta med frågorna efter konstruktionen av det konceptuella ramverket eftersom vi tror att det passar bättre att först strukturera upp de olika delarna i undersökningen, och deras inbördes relationer, och sedan fundera kring vad det är för frågor vi behöver ha svar på för att kunna skapa oss en trovärdig bild av undersökningsfenomenet, i vårt fall den underliggande systemutvecklingsmodellen och de medföljande nackdelarna för arbetet och i förlängningen resultatet. Viktigt att betona är hur som helst att det konceptuella ramverket och undersökningsfrågorna går hand i hand. (Miles & Huberman, 1994)

Miles och Huberman (1994) skriver att man ska hålla sig till ett mindre antal frågor, kanske som mest kring 10-12, gärna färre, för att man ska kunna arbeta med dem på ett givande sätt. Vi tror att de har

en klar poäng med sitt resonemang och anser vidare att det är särskilt viktigt för oss som oerfarna forskare som kan ha svårt att se samband och intressanta data om vi försöker greppa om ett för stort område. Vi har varit noga med att våra undersökningsfrågor ska vara nära relaterade till undersökningens syfte och vidare även sett till att de frågor vi fann relevanta gick att besvara på ett rimligt sätt. Det är ju ingen idé att ställa frågor till undersökningen som man av olika anledningar inte kan besvara. Nedan följer våra övergripande undersökningsfrågor:

- Vilken systemutvecklingsmodell hade man som utgångspunkt för utvecklingsarbetet?
- Hur påverkade systemutvecklingsmodellen utvecklingsarbetet?
- Hur nöjd var kunden med projektets resultat (Customer Satisfaction)?
- Hur väl höll projektet sig inom givna resursramar (Efficiency)?
- Vilka nackdelar innebar den valda systemutvecklingsmodellen?
- Vilka negativa effekter hade nackdelarna på Customer Satisfaction?
- Vilka negativa effekter hade nackdelarna på resursbehovet (Efficiency)?

2.1.3. *Undersökningsstrategi*

Den undersökning som ligger till grund för uppsatsen baseras naturligtvis på en vetenskaplig undersökningsstrategi för att forma arbetet. Det finns vissa moment i undersökningen som pekar på att vi kanske borde ha använt oss av någon form av kvasiexperimentell undersökningsstrategi då vi undersöker hur någonting förhåller sig efter att det så att säga har inträffat, oberoende av många yttre variabler i kontexten som organisationskultur och liknande. Vi undersökte ju hur det förhöll sig efter det att utvecklingsprojektet är genomfört och resultatet används (i de fall man inte valde att på grund av den klena kvalitén lägga ned projektet innan resultatet var ett faktum). Vi anser dock inte att vi borde ha använt oss av en sådan strategi då det inte fanns några förändringar i specifika oberoende variabler som vi var intresserade av, vi har ju bara undersökt hur det förhöll sig för att komma fram till vilka nackdelar som fanns i och med den sekventiella systemutvecklingsmodellen. (Svenning, 1999)

Istället har vi använt oss av en strategi av fallstudiekaraktär, i vilken vi ser de olika projekten som delar i en ”embedded single-case design” (Yin, 2003) vars resultat sammanställts i slutet av uppsatsen. I våra resultat försöker vi visa på de nackdelar som den sekventiella modellen innebär. Vilka nackdelar måste man ta hänsyn till med en sekventiell modell som grund för arbetet? Detta resonemang kommer vi att underbygga med hjälp av vår undersökning medelst analytisk generaliserbarhet. (Yin, 2003)

Viktigt för läsaren av uppsatsen är att veta att vi har vissa tidigare kontakter med de projekt som vi har valt ut. Det ger fördelar, som en ökad insyn samt en ökad samarbetsvilja hos de personer som undersöks. Det kan såklart, i och med att det kommer att präglå vårt arbete, även resultera i vissa nackdelar, som till exempel risken för en bristande construct validity (Yin, 2003) då vi påverkats i allt för stor utsträckning av våra tidigare etablerade uppfattningar och erfarenheter, trots att vi självklart försökt att minimera detta. Genom att ha varit medvetna om både fördelar och nackdelar hoppas vi kunna presentera ett så trovärdigt resultat som möjligt. Vi tar upp konkreta åtgärder för att motverka de negativa aspekterna senare i arbetet.

2.1.4. *Urval*

Definitionen av urvalet är någonting som kommer till stor nytta senare i analysen. Det handlar om det faktum att man inte kan undersöka alla aspekter av det område man intresserar sig för. Man begränsas av den tid och de resurser som man har till förfogande, och det är viktigt att man tar hänsyn till detta. De val man gör, vem man väljer att prata med, var man gör sin undersökning, när, hur och varför begränsar de slutsatser man kan dra ur undersökningen. Urvalet man gör handlar just om att göra dessa val explicita och tydliga. (Miles & Huberman, 1994)

Inom undersökningen har vi tittat på de entiteter som vi tog upp i det konceptuella ramverket ovan. Konceptuella ramverk och undersökningsfrågor sätter i stor utsträckning fokus och gränser för undersökningens urval, enligt Miles och Huberman (1994). Det finns dock ett par saker som vi tycker är värda att förtydliga. För det första måste man som läsare vara medveten om att vårt urval inte på något sätt varit hugget i sten, bara för att vi specificerade en utgångspunkt redan innan studien. Det görs för att explicit visa på den inriktning som vi tror att vi kommer att använda oss av i vår undersökning. Det kan ju som sagt, vilket också skedde, förändras vartefter som undersökningen visar på andra möjligheter som vi inte kunnat förutsäga, som kan komma att göra undersökningen mer fruktbar. Speciellt med tanke på att vi har undersökt tre fall. För det andra finns det såklart delar i vårt urval som inte explicit visas på enbart med hjälp av vårt konceptuella ramverk och våra undersökningsfrågor. Därav detta avsnitt.

När man som i vår undersökning planerar undersöka flera olika fall gäller det att man har ett tydligt ramverk för hur man väljer ut de, i vårt fall tre, olika fall som skall studeras. Här kan man välja olika strategier och vi har valt att välja ut de fall som vi ska använda ur projekt i vilka man kan se att man arbetat efter en sekventiell modell med tvivelaktiga resultat i form av förseningar, överskridna budgetramar osv. Vad vi strävade efter var alltså att använda oss av en allmän strategi där vi ställde upp ett antal kriterier som ett systemutvecklingsprojekt måste uppfylla för att vara intressant för vår studie, förutom att det måste falla inom definitionen för ett systemutvecklingsprojekt som vi ställde upp i kapitel 1. Miles och Huberman (1994) kallar denna strategi för Criterion.

De kriterier som vi ställde upp för att välja ut fall var följande:

- Det ska vara ett utvecklingsprojekt utfört inom de senaste fem åren för att man fortfarande ska ha en chans att få tag i relevanta personer och data, samt att respondenterna ska kunna komma ihåg projektet.
- Projektet ska minst ha haft fem och max tio deltagare förutom deltagare från kund. Detta för att inte projektet ska ha varit för litet eller för stort för vår undersökning. Det ska finnas en chans att få tag i relevanta personer (som ju ökar med personernas antal) men ändå skall projektet inte vara större än att vi kan greppa det inom ramarna för vår undersökning.
- Minst en av nyckelpersonerna i projektet, såsom projektledaren, ska kunna delta i studien.
- Projektet ska ha arbetat efter en sekventiell systemutvecklingsmodell
- Projektets resultat ska vara bristande i något avseende för att man lättare ska kunna hitta eventuella negativa påverkansfaktorer, alltså nackdelar, som systemutvecklingsmodellen fört med sig.
- Projektets ska omfatta traditionellt systemutvecklingsarbete enligt avsnitt 1.6.

Förutom att vi valde ut fall som tydligt uppfyllde de egenskaper som vi letade efter letade vi även efter fall som var någorlunda snarlika för att vi skulle kunna jämföra resultaten fallen emellan. Detta ledde till att vi kände att vi kunde specificera vad vi skulle undersöka i varje enskilt fall utan att behöva vara rädda för att dessa entiteter sedan endast skulle återfinnas i ett eller två av fallen. Vidare så var projekten lokaliserade i Sverige och nyckelpersoner fanns tillgängliga kring Lund/Malmö eller Stockholm eftersom detta underlättade för vår studie som inte hade alltför vida resursramar.

Vi var i varje enskilt fall intresserade av nyckelpersoner och då främst de som varit direkt involverade i större delen av arbetet. Hittade vi i våra fall personer som på ett eller annat sätt varit ansvariga för, eller som haft en nyckelroll inom, utvecklingsarbetet så var de mer intressanta för vår studie eftersom de då kunde förväntas ha en djupare förståelse för modellen som legat till grund för arbetet och på så sätt ge oss data av högre kvalitet. Vad vi ville försöka nå genom vår undersökning var ju en bild av de nackdelar som en sekventiell systemutvecklingsmodell kan föra med sig, och därför var detta naturligtvis av stort intresse.

Vi tog inte hänsyn till något statistiskt urval av undersökningspersoner utan istället försökte vi som vi nämner ovan finna de personer som kunde ses inneha de erfarenheter som vi var intresserade av. Detta eftersom vi inte är intresserade av någon form av statistisk generaliserbarhet, utan snarare av den analytiska generaliserbarhet som Yin (2003) talar om.

För att få tag i fall att arbeta med kontaktade vi nuvarande såväl som tidigare arbetskamrater och deras kollegor och bad om hjälp. Utfallet av vårt urvalsarbete blev tre systemutvecklingsprojekt som alla hade någon form av sekventiell utvecklingsmodell som grund för arbetet. De hade även alla bristande resultat i någon form och trots detta nyckelpersoner som var villiga att delta i studien. Vi var i början oroliga att vi inte skulle lyckas hitta tre bra fall, men det visade sig vara ett mindre problem än vi fruktat. De tre fall vi valt ut beskrivs närmare i kapitel fyra nedan.

2.2. Datainsamling

Nu när vi gått igenom vad det är vi velat undersöka i vår studie, och varför, är det dags att börja redogöra för hur vi har gått till väga. Yin (2003) menar att en nödvändig del av en fallstudie som tittar på flera fall eller analysenheter, är ett så kallat fallstudieprotokoll ("Case Study Protocol" (Yin, 2003)). Han menar att detta protokoll bidrar till ett ökat fokus på vad det faktiskt är man vill undersöka, samt tvingar forskarna att angripa de eventuella problem som kan förekomma i studien.

Då vi utförde en fallstudie med ett antal olika analysenheter ansåg vi det vara lämpligt att följa Yins råd och vi använde oss därför av ett fallstudieprotokoll. Vi såg dessutom andra fördelar med att använda detta, till exempel att vi fick en enhetlig grund att stå på då vi utförde datainsamlingen i de olika enheterna. Vi visste att vi hela tiden åtminstone hade samma utgångspunkt vilket vi anser gav en bättre grund för analys och jämförelser mellan enheterna. Vi använde oss dock inte av ett så utförligt protokoll som Yin visar på, då vår undersökning var begränsad i omfattning samtidigt som all datainsamling gjordes av båda författarna i samarbete. Vi upplevde att det då inte fanns samma behov av en extensiv grund att ta med sig ut på plats, vi var båda väl insatta i vad det var vi ämnade studera. Fallstudieprotokollet återfinns i bilagorna.

2.2.1. *Instrumentation*

Enligt Miles och Huberman (1994) så innebär begreppet instrumentation att man definierar specifika metoder som man tänker använda för att samla in data. Man kan ange metoder beroende på om det är kvalitativ eller kvantitativ data man är intresserad av, samt antingen på ett löst hållet sätt eller hårdare.

I vår studie är vi intresserade av kvalitativa data vilket avspeglas i de metodval som redovisas nedan. Anledningen till att det är kvalitativa data som är av intresse i undersökningen är att vi avsåg att undersöka hur respondenterna upplevt att systemutvecklingsmodellen påverkat arbetet i projektet såväl som rena fakta. Det leder till att vi eftersökte ett djup som vi inte anser att vi kunde uppnå lika väl med kvantitativa data.

Miles & Huberman (1994) för en diskussion kring huruvida man bör använda sig av ett löst eller ett hårt hållet förhållningssätt i sin instrumentation. Vi är av åsikten att det i stor utsträckning beror på vilken typ av studie det är man ämnar genomföra. Vissa typer av studier, som multipla fall där kontexten är av mindre betydelse, tjänar på att ha en hårdare hållen instrumentation emedan andra, som där uppgiften är mer av en förklarande än en bekräftande typ, tjänar på en lösare instrumentation med mer utrymme för att avgöra vilka datainsamlingsmetoder som är användbara och vilka som ska användas, under själva studien. Som man kan se bör vår instrumentation hamna mer åt det hårdare hållet och vi har därför valt att ha en tämligen hårt hållen instrumentation under arbetet med till exempel verktyg som intervjuguiden. Detta även för att som forskare sätta fokus på de antaganden som vi anser att man gör kring fenomenet i en undersökning och göra dem explicita.

2.2.2. *Datainsamlingsmetoder*

I vårt datainsamlande har vi, som vi tidigare sagt, valt att bygga upp vår fallstudie på undersökande av tre olika projekt. Det finns dessutom två delar av varje fall; dels så har vi en beskrivande del där vi tar reda på information kring själva projektet som resultatet, systemutvecklingsmodellen man använt och hur många som involverats, dels så gäller det att få svar på resten av våra undersökningsfrågor.

Den första delen handlade alltså om att vi ville skapa oss en bild av projektet. Detta gjorde vi med hjälp av ostandardiserade och ostrukturerade intervjuer av en eller ett par deltagare. Anledningen att vi lämnade mycket fritt i intervjun med både en låg standardisering och en låg struktur är att vi inte var intresserade av någon form av möjlighet till generalisering av svar eller liknande, utan endast av att tolka deras svar och skapa den bild som vi nämner ovan. Tanken är att intervjuerna ska ta sig uttryck som samtal. På det sättet hoppas vi att respondenterna ska känna sig trygga i intervjun och att det ska återspeglas i djupet i, och kvaliteten på, deras svar.

I den andra delen genomfördes ostandardiserade semistrukturerade intervjuer av nyckelpersoner i projektarbetet. Vi har i skapandet av våra intervjufrågor varit noga med att följa en ”chain of evidence” (Yin, 2003) från syftet och frågeställningen via vårt konceptuella ramverk och våra undersökningsfrågor till våra intervjufrågor. Detta har vi gjort bland annat genom att utgå från våra undersökningsfrågor när vi skapade våra intervjufrågor och låtit varje grundfråga i intervjuguiden direkt motsvara en undersökningsfråga. Våra intervjufrågor återfinns i vårt fallstudieprotokoll i bilagorna.

Vidare så har vi i vår sammanställning av data från fallstudien varit noga med att låta alla resonemang direkt bygga på data vi fått fram i våra intervjuer. Sammanställningen av våra intervjuanteckningar återfinns i bilagorna. Denna sammanfattning ligger sedan till grund för kapitel 4 som i sin tur ligger till grund för vår analys och våra slutsatser. Detta så att vi kunnat vara säkra på att vi undersökt det som vi ämnade undersöka och att våra resonemang verkligen baseras på våra fall. Vi har alltså underbyggt vår undersöknings validitet (Yin, 2003).

För att motverka den påverkan som vi tog upp ovan med att vi har tidigare kontakt med respondenterna så har vi dels genom seminarier med handledare genomgått vad man skulle kunna kalla för en expert review (jmf Heuristic Evaluation inom området Human Computer Interaction (Löwgren 1993, Preece et al 2002)). Dessutom har vi under hela arbetet med att samlas in data använt oss av forskartriangulering (Woods et al, 2000) då vi båda närvarat vid intervjuerna. En av oss ställde frågorna och en förde anteckningar. Efter intervjuens genomförande gick vi igenom anteckningarna tillsammans för att kontrollera så att de avspeglade vad som sagts och vi stämde även omgående av våra anteckningar med respondenten. Genom dessa åtgärder upplever vi att vi kraftigt minskat risken för att undersökningens resultat ska påverkas negativt.

För att vi inte skulle få för enkelspåriga svar var vi noga med att inte bara intervjua en person, utan minst två per projekt. Det slutgiltiga antalet avgjordes under arbetets gång, då vi fick en uppfattning om de svar de olika respondenterna gav, hur många som var lämpliga att intervjua och så vidare.

Yin menar vidare att man bör utföra en pilotstudie för att testa kvaliteten på fallstudieprotokollet, på analysfrågorna och egentligen hela designen av fallstudien. Vi hade dock tyvärr relativt snäva tidsramar och således begränsade möjligheter att göra några djuplodande pilotstudier. Detta är ett problem för vår undersökning, då kvaliteten tyvärr riskerar att bli lidande på grund av detta. Vi är dock medvetna om denna brist och anser att vi motverkat den i hög grad genom att dels genom arbetet hålla en tydlig inriktning på undersökningen och vara noggranna och ingående i vårt arbete med att till exempel skapa vårt fallstudieprotokoll och dels genom att vi stämde av intervjufrågorna genom att intervjua en kollega kring ett helt annat projekt. Detta ledde till en mindre omarbetning av vissa intervjufrågor.

Vi har vidare inte bandat våra intervjuer, beroende av tre anledningar:

- Återkoppling till respondenten går snabbare om man inte bandar intervjun eftersom man direkt efter genomförd intervju kan gå igenom intervjuanteckningarna tillsammans med respondenterna. Detta passade oss utmärkt eftersom vi inte hade så mycket tid till vår undersökning samt eftersom det gör att respondenterna har fallet aktuellt direkt efter intervjun och då bättre kan avgöra intervjuresultatets kvalitet.
- Det var väldigt viktigt för vissa av våra respondenter att känna sig anonyma. För att nå en hög kvalitet på svar och dessutom från ett etiskt perspektiv var det viktigt att just dessa respondenter kände sig trygga. Kalle ville absolut inte att vi bandade dennes svar eftersom delar var väldigt känsliga för hans nuvarande arbetssituation. Detta ledde oss till att tro att eventuellt fler skulle känna likadant.
- Vi förväntade oss att det under intervjuerna skulle kunna komma fram mycket känsligt material, som utpekningar av medarbetare eller chefer med negativa anspelningar. Detta är lättare att bortse från när man tar anteckningar än när man gör en bandupptagning med påföljande transkribering.

Istället förde vi anteckningar med hjälp av laptop samt i två fall papper och penna för att skriva ned nyckelbegrepp och stödanteckningar för vidare arbete. Dessa lösa anteckningar sammanställdes sedan till likartat format. Genom att se till att alla data fanns i digital form och dessutom var grupperade efter intervjufrågor underlättades vårt arbete i analysen nedan.

För att få en högre grad av datatriangulering (Miles & Huberman, 1994) har vi även tagit del av de dokument som funnits rörande arbetet i projektet och den underliggande systemutvecklingsmodellen. Med dokument menar vi, i enlighet med Patel och Davidsson (1994), mer än bara tryckt material. Annan media som kunnat ge stöd till undersökningen var såklart också av intresse. Dokumenten har dock setts som ett komplement då det är hos intervjuvaren som tyngden legat.

2.3. Analys av data

För att påbörja analysen av de data man samlat in måste man först och främst koda den (Miles & Huberman, 1994). Enligt Miles och Huberman (ibid.) så är kodning ett sätt att differentiera och kombinera data samtidigt som man reflekterar över informationen. Det är precis vad vi ville göra med vår kodning och vi anser att det är bra att som forskare vara medveten om detta.

Trots att undersökningen har mycket tydliga mål och ett klart undersökningsområde ansåg vi det vara motiverat att inte på förhand helt färdigställa de koder som vi använde i dataanalysens inledningsfas. Detta motiverar vi med att vi hade mycket svårt att i början av undersökningen veta allt om vad vi skulle komma att se i våra data. Vi kunde dock identifiera vissa nyckelbegrepp och kategorier för koder redan tidigt och därför valde vi att göra ett utkast på koder innan vi påbörjade insamling av data, som vi dock visste skulle komma att förändras. I vårt utkast utgick vi från vårt konceptuella ramverk samt våra undersökningsfrågor som ju sätter ramarna för vår studie. De entiteter som vi hittade i det konceptuella ramverket gjorde vi till generella kategorier för koder.

När vi samlat in våra data och sammanställt våra anteckningar så inledde vi vår kodning med att färdigställa den mall för koder som vi ämnade använda oss av. Detta arbete utgick i vår medvetenhet om karaktären på de data vi samlat in. I de fall vi upplevde att det fanns behov av en ytterligare kategorisering skapade vi koder för det. Mallen över de koder vi använt återfinns i bilagorna. Där finns även två exempel på hur vi kodade våra data.

Under arbetet med kodningen så skrev vi regelbundet ned intressanta idéer kring relationer eller intressanta uppslag som uppenbarade sig. Detta gjorde vi dels för att minnas de intressanta uppslag som vi inte hann följa upp under kodningen samt för att binda samman olika delar av all data till mer överskådliga kluster. Dessa ”Memos” skrevs som en hjälp till analysarbetet på post-it-lappar som sedan grupperades på en whiteboard. (Miles och Huberman, 1994)

Då vi genomförde vår analys utgick vi först och främst från en konceptuellt ordnad matris där vi ställde upp våra undersökningsfrågor i den vertikala dimensionen och relaterad data från våra fall i den horisontella. På detta sätt kunde vi också jämföra resultat över fallgränserna för att på så sätt få ett tydligare resultat.

Efter det att vi analyserat vart och ett av de olika projekten, gick vi vidare för att försöka ställa de respektive fallens resultat mot varandra för att kunna se likheter eller skillnader dem emellan. Här utgick vi från de matriser vi skapat för att skapa en enklare variant av ett kausalt nätverk (Miles och Huberman, 1994) för att visa på en gemensam grund för de olika fallen.

2.4. Etik och kvalitet i undersökningen

Vi har under vårt arbete haft en klar ansats att uppnå vetenskaplig och etisk kvalitet. Vi har använt oss av ett etiskt ramverk, i vilket vi behandlade etiska frågor som var viktiga för undersökningen. Eftersom vi ser en likhet mellan etik och krav på god kvalitet så innefattade vårt etiska ramverk både den vetenskapliga och den etiska kvaliteten. (Wallén, 1996) Miles & Huberman (1994) anser att det är viktigt att man som forskare tar ansvar för att ens resultat inte är triviala, att de bidrar till någonting större än ens egen karriär eller liknande vilket vi håller med om.

2.4.1. Etiskt ramverk

Till att börja med kan det vara värt att påtala att undersökningen utförts utan någon ekonomisk vinning för vår egen del. Det har inte förekommit någon finansiering av arbetet och på så vis behöver man som läsare inte oro sig för att vi skulle ha påverkats av de krav som våra finansörer i annat fall skulle ha ställt på våra resultat. (Wallén, 1996) Vi ser därför att våra resultat ägs av oss och det är vi som har rätt att bestämma hur de ska användas. På så sätt vill vi kunna garantera respondenternas säkerhet i så stor utsträckning som möjligt.

Vi har under vårt arbete iakttagit mycket stor försiktighet kring våra respondenter och varit noggranna med att de inte ska påverkas negativt av undersökningen. Allmänt i forskning gäller oftast att individen ska ha gett sitt ”informerade samtycke”, vilket vi har sett till i alla fall. Individen har även kunna avbryta sitt deltagande i studien när som helst. Inga potentiellt skadliga resultat går att relatera till en specifik individ eller grupp bland undersökningspersonerna. (Patel & Davidson, 1994; Wallén, 1996) Detta blir särskilt relevant i vår typ av studie där vi ju undersöker problem och känsliga situationer. Om en respondent pekar ut en kollega som ansvarig för att en nackdel fått svåra konsekvenser ser vi det som mycket viktigt att varken kollegan eller respondenten går att identifiera.

En insamlingsmetod vi använt oss av är ostrukturerade intervjuer, och den tål att tittas närmare på. Detta är en väldigt subjektiv teknik i vilken risken för skevhet är stor. Det kan även visa sig svårt att analysera de svar man får då de är komplicerade, innehållsrika och varierade. Detta kan påverka resultatens kvalitet. Det finns förutom det alltid en risk att man får en viss skevhet i resultaten genom att man som intervjuare påverkar respondenterna under det att man genomför intervjun. Vi anser att vi genom att vi varit medvetna om detta och gjort allt för att minimera problemet och förbättra kvaliteten har klarat oss undan de negativa effekterna. (Bell, 1995; Patel & Davidson, 1994)

Slutligen vill vi ta upp kritisk källanalys som är avgörande för kvaliteten i analysen. Genom att ha både ett externt och ett internt kritiskt förhållningssätt till våra datakällor så har vi nått en högre kvalitet i våra resultat. Hänvisningar till källor har gjorts med ett kritiskt förhållningssätt, så att informationen inte tas ur sitt sammanhang eller ges mer vikt än vad som kan vara rimligt. (Bell, 1995)

2.4.2. *Validitet*

Att ha en hög validitet i undersökningen är viktigt eftersom det innebär att de metoder man använder undersöker rätt saker och att de data man får fram reflekterar verkligheten. Vi anser att eftersom vi använt oss av mer än en respondent per fall och dessutom styrkt våra primärdata (intervjuer) med sekundärdata (dokumentation) har vi goda möjligheter att uppnå hög validitet. Detta är ingenting som vi klart kan bevisa, undersökningen baserar sig ju på en subjektiv bedömning av verkligheten, men då vi låtit våra respondenter ta del av vårt arbete och fått den uppfattningen att de ställer sig bakom våra data och våra resultat vill vi mena att vi lyckats. (Yin, 2003)

2.4.3. *Reliabilitet*

Ett annat viktigt begrepp att ta hänsyn till för att säkerställa kvaliteten är reliabilitet. En hög reliabilitet innebär att en annan undersökare skulle kunna upprepa vår studie och då komma fram till samma resultat. För att öka vår reliabilitet har vi nogsamt dokumenterat vårt tillvägagångssätt i detta kapitel samt bilagt en sammanställning av de data vi samlat in i intervjuer, våra intervjuanteckningar, i bilagorna. Vi har även bilagt de koder vi använt oss av i analysen. Med detta stöd anser vi att en annan forskare troligen skulle kunna göra en liknande undersökning och komma fram till likartade resultat.

2.5. Rapporten

En rapport ska skrivas till dem som ska läsa den. För att man ska kunna producera en rapport med så hög kvalitet som möjligt så måste man börja med den i tid. Det lönar sig att redan tidigt i studien börja fundera över en del begrepp som hör till rapporten.

Det är alltså viktigt att man skriver på ett sätt som kan ge önskad effekt hos dem som läser rapporten. För oss handlar detta om att försöka bidra till en ökad medvetenhet bland läsare av rapporten, förhoppningsvis morgondagens systemvetare, om just de nackdelar som utvecklingsarbete baserat på en sekventiell systemutvecklingsmodell kan föra med sig. Vi tror att genom att visa på de resultat vi kommer fram till, oavsett deras karaktär, kan vi få de personer som någon gång i framtiden kommer att arbeta i systemutvecklingsprojekt att fundera över de resultat som vi visar på, tillgodogöra sig dem och väga in denna kunskap i sitt arbete. Detta är vårt kunskapsbidrag.

Läsarna kan med andra ord förväntas ha samma akademiska nivå som vi har, vilket vi tar hänsyn till i rapporten. Vidare är det viktigt att man tar hänsyn till det språk man vill ska prägla rapporten, vilket i sin tur ligger till grund för den relation man vill bygga upp mellan sig själv som forskare och läsaren. Vilken roll vill man att läsaren ska uppfatta att man har? Vi känner att den troligaste rollen för oss att förmedla till läsaren blir den som undersökande, nyfiken kollega som visar på hur det fungerar i ett antal fall och på de nackdelar som sekventiella modeller kan medföra. (Miles & Huberman, 1994)

Vidare vad gäller språk och struktur så har vi försökt hålla oss på en akademisk nivå då våra presumtiva läsare kan förväntas ha samma förkunskaper som vi vad gäller ordförråd och begrepp. Vi har

dock inte försökt oss på avancerade strukturer som poem eller liknande, utan har hållit oss till vanlig förklarande och beskrivande text, med en linjär disposition. Stilmässigt har vi försökt hålla texten välskriven. Vi har kompletterat texten med bilder, figurer och matriser där det varit lämpligt för att öka läsbarheten och göra vårt arbete och våra resultat lättare att ta till sig.

3. Systemutvecklingsmodeller och -processer

I detta avsnitt kommer vi att beskriva ett antal olika systemutvecklingsmodeller och hur utvecklingen har sett ut inom detta område de senaste decennierna. Denna beskrivning kommer i mångt och mycket att utgå från sekventiella modeller kontra evolutionära modeller.

Vidare kommer vi, förutom att titta på systemutvecklingsmodellerna, även att titta på ett antal olika systemutvecklingsprocesser som faller inom ramen för de olika modellerna för att vi vill visa hur modellernas koncept praktiseras. Detta gör vi också för att visa på att det faktum att modellerna om-sätts till mer praktiskt användbara processer inte gör att svagheter i dessa underliggande modeller försvinner.

Avslutningsvis kommer vi att sammanställa ett flertal olika författares kritik mot de sekventiella modellerna för att på så vis få en lista över nackdelar med dessa modeller, och även titta på hur dessa nackdelar hanteras av de evolutionära modellerna.

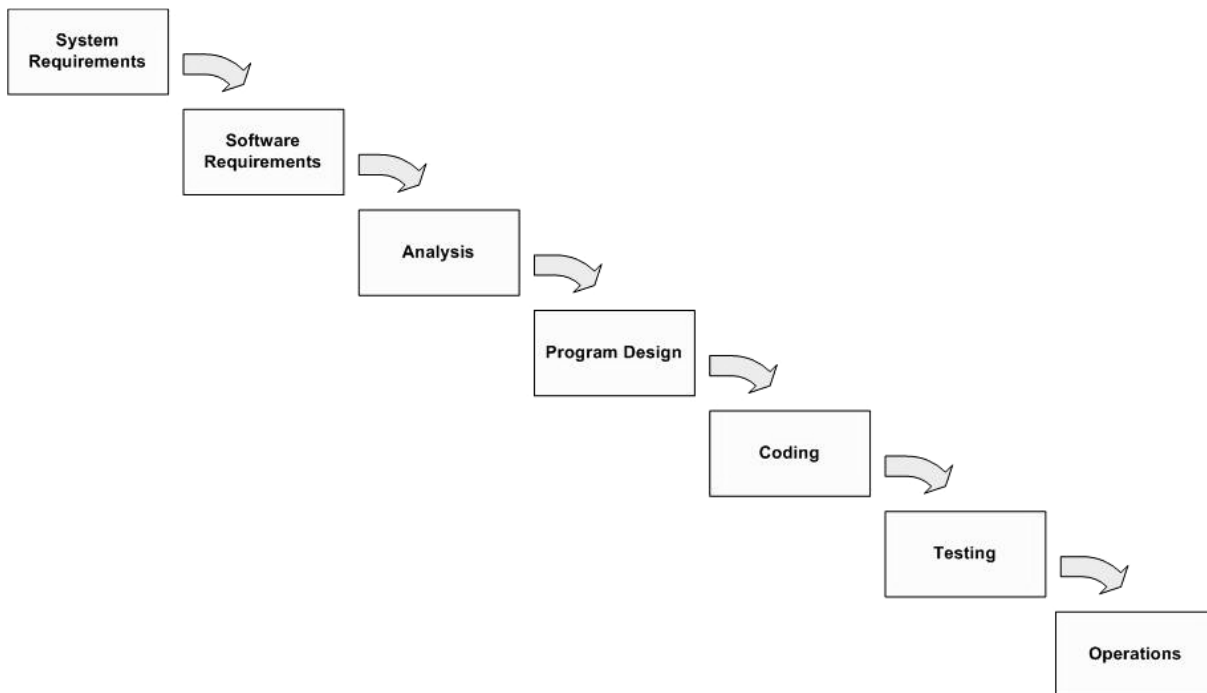
Grundläggande för alla systemutvecklingsprocesser är som vi sagt tidigare att de har sin utgångspunkt någonstans i ett spektrum av systemutvecklingsmodeller för hur ett systemutvecklingsprojekt egentligen genomförs från start till slut. Royce (1987) menar att en av de första av dessa systemutvecklingsmodeller var ”koda-och-fixa”-modellen. I denna gör man mycket lite eller ingen design alls, ingen kravspecifikation, inga tester, utan enbart en kortare analys och därefter kodning. Uppstår några problem fixar man dessa med lite mer kodning. Detta leder enligt Gulliksen & Göransson (2002) till överskådlighet i systemet och en stor klyfta mellan användarnas önskemål och systemets funktionalitet. Royce (1987) menar dock att denna modell ofta är utgångspunkten om det system som utvecklas ska användas av de personer som utvecklat det. I övriga fall är ett projekt med denna utgångspunkt dömt att misslyckas (Royce, 1987).

3.1. Sekventiella modeller

Då man någon gång under 50-talet upptäckte de stora brister som en koda-och-fixa-modell leder till, uppstod ett behov av mer strukturerade modeller för systemutveckling och de första sekventiella systemutvecklingsmodellerna uppstod (Boehm, 1988). En sekventiell modell är en modell där man har ett antal faser som går igenom allt eftersom projektet fortskrider (Kruchten, 2002). En fas syftar till att utföra en del i projektet som framtida faser sedan kommer att bygga vidare på eller använda i arbetet med just den fasan. Ett exempel kan vara den kravspecifikation som skapas i en fas och som sedan ska ligga till grund för designarbetet med att ta fram en systemspecifikation. Denna ligger i sin tur sedan till grund för konstruktionsarbetet. Faserna är som namnet antyder helt sekventiella; man kan inte arbeta med två faser samtidigt, utan en fas påbörjas först när föregående fas är helt slutförd.

Den mest grundläggande av alla dessa modeller är en av dem som Boehm (1988) talar om; den stegvisa modellen. I denna går man igenom utvecklingsprojektet steg för steg, fullständigt sekventiellt. Man kan inte gå till steg tre från steg ett utan att passera steg två. Inte heller kan man backa tillbaka i projektet för att göra om något steg eller komplettera, vilket i teorin skulle ställa extremt höga krav på projektdeltagarna att göra ett bra och ingående arbete i ett steg eftersom nästföljande steg bygger på det föregående (Boehm, 1988). Denna modell kallas ofta också för livscykelmodellen och består av ett antal olika faser som ska gås igenom. Styrkan i denna modell är för det första dess enkelhet; det är extremt enkelt att förstå hur projektarbetet kommer att förlöpa om man väljer att använda en utgångspunkt som det här. För det andra var detta det första steget som togs mot att försöka höja abstraktionsnivån i utvecklingen för att få en systemarkitektur som bland annat underlättar vidareutveckling och underhåll. (ibid.)

I projektarbetet utförs som tidigare sagt ett antal olika faser som olika författare delar upp på olika sätt. Royce (1987) menar att man utför sju faser som finns illustrerade i figuren nedan.



Figur 3-1: Den stegvisa modellen (egen konstruktion)

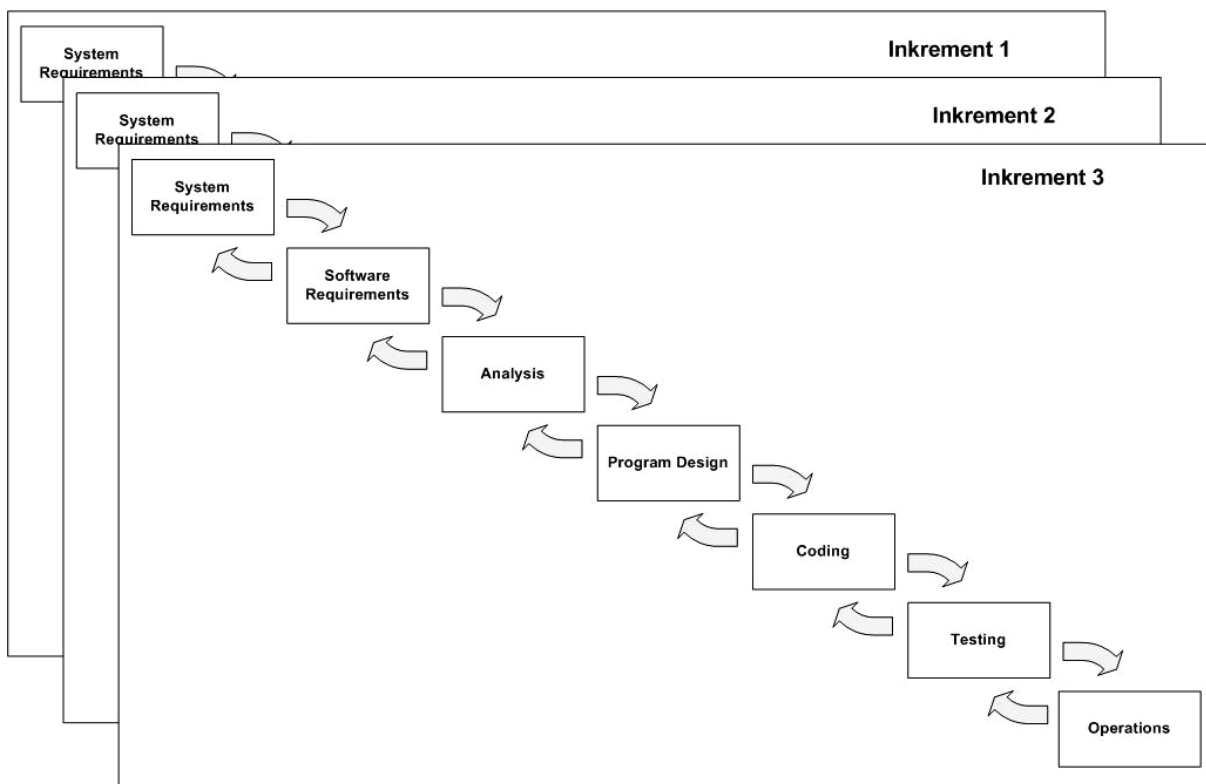
Denna stegvisa modell har genom åren fått mycket kritik, och har ofta blandats ihop med en variant av densamma, den så kallade vattenfallsmodellen. Detta är egentligen en version av den stegvisa modellen där man för det första tillåter ”backning” till föregående fas, och även medger att prototyping kan vara ett lämpligt verktyg att använda (Boehm, 1988).

Detta sätt att arbeta förutsätter att man använder sig av en riklig dokumentation av ett antal olika anledningar. För det första så är det viktigt att alla systemdesigners har samma utgångspunkt och att man har allt på pränt så att designen är tydlig. Med designen menas här själva dokumentationen då denna faktiskt *är* både designen och specifikationen, eftersom detta är det enda man har innan man börjat konstruera själva systemet. På samma sätt är det viktigt att såväl i det fortsatta arbetet i projektet som i det framtida underhållet och eventuell vidareutveckling viktigt att kunna gå tillbaka till den-

na dokumentation för att klargöra specifika designval, problem och särskilda krav. Observera här att det inte handlar om någon återkoppling till tidigare faser där man kan revidera sin dokumentation, utan det som är gjort i analysfasen står fast när man väl kommit så långt att man börjat konstruera systemet. Här handlar det helt enkelt om att man ska kunna studera den dokumentation man genererat i tidigare faser.

För att hantera en ökad komplexitet i systemen man utvecklar, kan man använda sig av en inkrementell modell, vilket är ytterligare en variant på vattenfallsmodellen. I detta fall delar man upp hela systemet i ett antal mindre delar där varje del utvecklas för sig själv, i ett eget litet projekt, med en helt egen livscykel. Man kan antingen utveckla de olika delarna sekventiellt, alltså efter varandra, eller parallellt. En uppdelning på detta vis ger en bättre överblick, bättre samstämmighet med användarnas krav och en ökad utvecklingshastighet. (Boehm, 1988 samt Gulliksen & Göransson, 2002)

Vanligtvis används vattenfallsmodellen som utgångspunkt i varje enskild del av systemet, vilket kan få betydligt bättre resultat här än med vattenfallsmodellen som utgångspunkt för hela systemet. Detta på grund av att riskerna som finns med att till exempel brister i designen upptäcks i testfasen får betydligt mindre konsekvenser, då bristerna omfattar betydligt mindre delar. (Boehm, 1988)



Figur 3-2: Inkrementell modell (egen konstruktion)

Trots att denna modell i sina inkrement består av traditionella vattenfallsmodeller, finns här en skillnad i och med att man tillåts arbeta med de olika delarna parallellt. Man börjar alltså försöka ta sig ifrån det sekventiella angreppssättet, men har fortfarande kvar många av de problem som föreligger med de sekventiella modellerna.

3.1.1. Exempel på sekventiella systemutvecklingsprocesser

Det finns många systemutvecklingsprocesser som grundar sig i ett sekventiellt tänkande. Bland dessa märks till exempel *Strukturerad Analys och Konstruktion av informationsbehandlingsystem (SAK)* och *Jackson System Development (JSD)*, vilka är två processer som förekommer regelbundet i litteraturen. Nedan kommer vi att beskriva den senare av dessa för att visa ett exempel på hur de sekventiella modellerna återspeglas i specifika processer.

Grundinställningen i JSD är att alla system endast är en modell av verkligheten (Rollo, 1992). Ett banksystem skulle till exempel vara en spegling av själva banken komplett med kunder, bankkonton och pengar. Processen är uppbyggd i huvudsak av tre faser som utförs sekventiellt i tur och ordning, vilka i stort sett kan ses som de tre faserna *Analys*, *Design* och *Konstruktion*:

- **Modelling Phase**

I denna fas försöker man återskapa och åskådliggöra den önskade verkligheten med hjälp av en modell. I denna modell fokuserar man på de delar av verkligheten som är relevanta för systemet och bortser helt från övriga delar.

Modellerna byggs upp med utgångspunkt från de aktiviteter som varje modellerad entitet utför. Ett exempel från bankvärlden kan här vara entiteten *Kund* som utför aktiviteten *Sätter in pengar* på entiteten *Bankkonto*. Varje aktivitet beskrivs i modellerna som innehavare av ett antal olika egenskaper. Till exempel skulle insättningsaktiviteten ha egenskaper såsom tidpunkt och belopp. (Rollo, 1992)

- **Specification Phase**

I denna fas skapar man en ingående systemspecifikation över hela systemet där man först och främst fokuserar på att komplettera det modellerade systemet med funktioner för att få systemet att fungera. Man skapar också ett "systemnätverk" genom att dela upp systemet i mindre delar och tydligt specificera hur dessa delar ska interagera med varandra. Denna uppdelning till mindre delar innebär att man rör sig mot en utgångspunkt i den inkrementella modellen. (Rollo, 1992)

- **Implementation Phase**

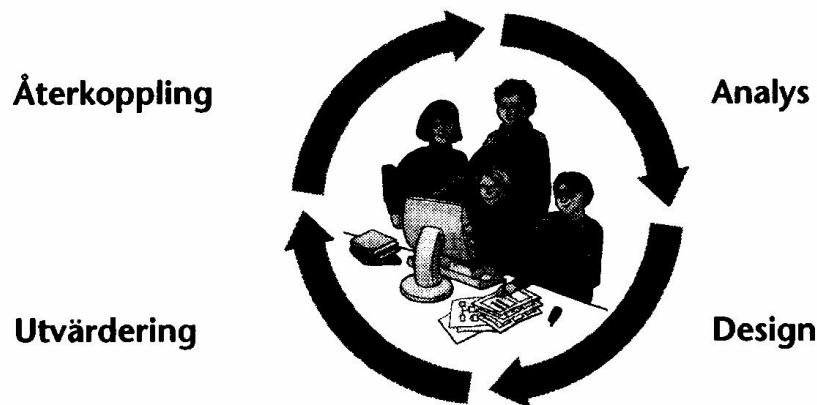
I denna fas omsätts den systemspecifikation man tidigare framställt till ett fullt fungerande system. Utgångspunkten är att man utgår från de ingående krav- och systemspecifikationer man tidigare satt upp och transformerar dessa till att passa målmiljön med hård- och mjukvara. Denna transformation sker genom att man på förhand skapar en ingående specifikation över vilka transformationer som krävs och hur dessa ska utföras. (Rollo, 1992)

3.2. Evolutionära modeller

Den grundläggande skillnaden mellan de sekventiella modellerna och evolutionära modeller är att de senare inte har några sekventiella faser. Istället arbetar man mer iterativt för att på så sätt avancera i utvecklingsarbetet. Allt eftersom tiden går får man således mer och mer kunskap om det aktuella

problem och kommer hela tiden närmare en passande lösning. Man har alltså möjlighet att angripa komplexa problem som i inledningsskedet av projektet kan vara mycket svåra att identifiera och analysera. I illustrationen nedan visas att återkopplingen till analys och design är det som är centralt i denna modell. (Gulliksen & Göransson, 2002)

Här kan man se hur man i och med att man gör samma sak flera gånger iterativt har börjat inse att det är lämpligt att återkomma till exempel design regelbundet, vilket indikerar på ett behov av ett mer parallellt arbetssätt mellan olika aktiviteter, som i de sekventiella modellerna delas upp i faser. (ibid)



Figur 3-4: Iterativ utveckling (Gulliksen & Göransson, 2002 s.146)

3.2.3. Exempel på evolutionära systemutvecklingsprocesser

Även bland de evolutionära modellerna finns mängder av olika systemutvecklingsprocesser. Bland de största här märks till exempel Extreme Programming (XP) och Dynamic System Development Method (DSDM). I båda dessa har man möjlighet att röra sig fritt mellan olika delar av projektarbetet allt ifrån analys och design till konstruktion. För att även här visa hur en process kan använda sig av en evolutionär modell i sin konstruktion så kommer vi att beskriva XP relativt övergripande.

XP:s utvecklingsprocess börjar med att utvecklingsprojektets medlemmar tillsammans med kunden skapar en översiktlig kravspecifikation, genom att bland annat analysera den rådande samt den önskade situationen. Metoden utgår från att bygga upp hela systemet i mycket små delar, vilka ska utvecklas under mycket korta iterationer, som lämpligen fullföljs under tvåveckorsperioder. Efter att varje del har blivit levererad utvärderar användarna systemet och kommer med feedback till utvecklarna huruvida detta verkligen var vad de ville ha. När användarna är nöjda med en del, går utvecklingen vidare med nästa lilla komponent. (Beck, 2001)

Utvecklingsprocessen består enligt Beck (2001, s. 86) i huvudsak av två steg: Planering (som namnet till trots innehåller mycket mer än planering) och Iterationer.

Utvecklingsprocessen i XP inleds alltså med att man gör en övergripande planering av projektet. Utifrån scenarion som man här bygger upp preciserar hela projektgruppen gemensamt vilka problem som ska lösas och rangordnar dessa i den ordning de bör genomföras, samt gör en uppskattning om ungefärlig kostnad och tidsåtgång. Viktigt att lägga märke till är att representanter från kunden måste

ingå i projektgruppen, då utvecklarna av XP menar att dessa är de enda som verkligen är insatta i verksamheten som avses. (Beck, 2001)

Då projektgruppen lagt upp planen för systemets olika delar och hur viktiga de är, inleds själva design- och utvecklingsarbetet. Detta arbete sker som tidigare har nämnts genom korta iterationer, där den del som ansågs viktigast i den övergripande planeringen är den som först utförs. Övriga delar kommer till viss del att utgå ifrån den första iterationen, och detaljplanering av deras respektive uppgifter kommer således först senare, i respektive iteration. Utvecklingsarbetet ska präglas av att de nya komponenter man skapar i iterationerna ska integreras sömlöst med de övriga delarna. (Beck, 2001)

3.3. Nackdelar med de sekventiella modellerna

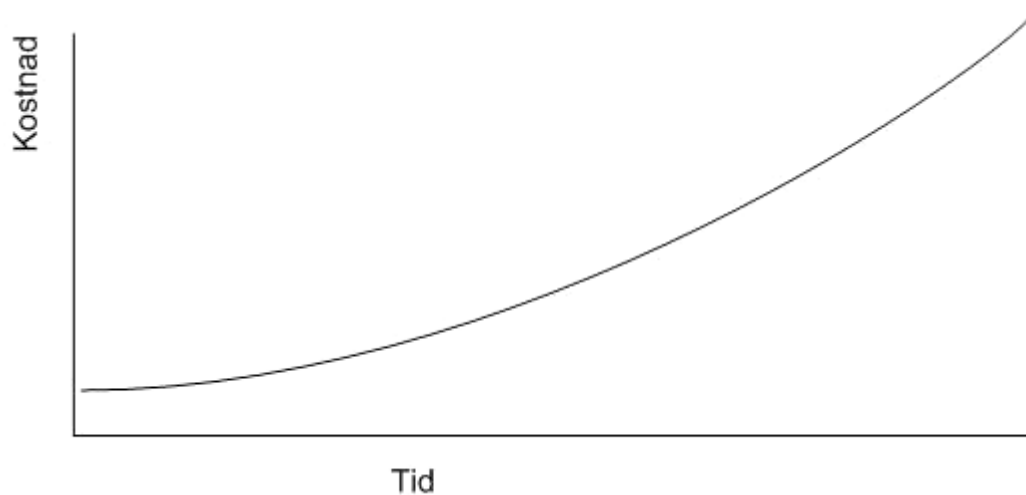
För att visa på vilka nackdelar som eventuellt finns med de sekventiella modellerna, så har vi valt att undersöka olika författares syn på dessa modeller för att på så sätt ta fram de nackdelar och den kritik som de visar på. Något som vi hittat hos ett flertal olika författare är att man menar att de sekventiella modellerna har mycket svårt att hantera förändringar (Exempelvis Kruchten, 2002, Beck, 2001 och Boehm, 1988). Detta anser vi vara en av de största nackdelarna med de sekventiella modellerna då förändring är en så pass stor del av ett projekt (Dvir & Lechler, 2003). Här ser vi dessutom två olika grupper av förändringar. För det första ser vi att förändringar i kraven är ett stort riskmoment, och för det andra kan många andra faktorer också påverka, däribland tillgången till resurser eller förändringar i tekniska förutsättningar.

Vidare kan vi se att en annan nackdel som tas upp i litteraturen är att ett projekt som arbetar sekventiellt ofta är primärt fokuserad på att producera dokument och specifikationer och inte ett informationssystem. Vilket dessutom hänger nära samman med att vissa författare talar om att sekventiella projekt är allt för tekniskt inriktade och tenderar att avskärma användarna från utvecklingsprocessen.

Dessa fyra kategorier är de vi anser vara de mest omskrivna i litteratur och forskning, vilket gör att vi kommer att utgå från dessa i vårt fortsatta arbete. Nedan kommer vi att ta upp dessa kategorier en efter en och redogöra mer ingående för dem.

3.3.1. Förändringar i systemkraven

Kruchten (2002) talar som sagt om att ett av de mest grundläggande problemen med de sekventiella modellerna är att denna gör det felaktiga antagandet att man i början av projektet kan specificera en komplett lista över alla de krav som ställs på systemet. Grundförutsättningen i vattenfallsmodellen är att man måste se till att göra det genomarbetat eftersom man inte har särskilt stora möjligheter att förändra detta, vilket Kruchten dock menar inte alls är genomförbart. Han menar att *”hur mycket vi anstränger oss kommer detta nästan alltid visa sig vara omöjligt”* (Kruchten, 2002, s. 55), eftersom kraven alltid har en tendens att förändras. Här menar Beck (2001) att det stora problemet är att förändringar dessutom tenderar att bli dyrare att genomföra ju längre man nått i ett systemutvecklingsprojekt med utgångspunkt i ett sekventiellt tänkande. Man skulle kunna visa en graf som visar kostnad för förändringen vid olika tidpunkter i projektet.



Figur 3-5: Kostnad för förändring i sekventiella modeller

Denna graf visar att kostnaden för att hantera förändringar stiger exponentiellt allteftersom projektet fortskrider. Vilka förändringar är det då som Beck och Kruchten talar om här? Kruchten (2002) menar egentligen att några av de faktorer relaterade till systemkraven som kan förändras är:

- Användarnas behov

Användarnas behov kan komma att förändras under tiden som utvecklingsförloppet går. Detta kan särskilt gälla om man har projekt som löper över flera år istället för månader. Förändringarna kan till exempel grunda sig i förändringar i användarnas arbetsmiljö som eventuellt kan komma att påverka deras behov, men även faktorer som deras kunskapsnivå som kanske höjs då man arbetar med projektet, då man till exempel ser andra system och dessas möjligheter och begränsningar.

- Problemet

Vad användarna tror att de vill ha och vad de vill ha när de ser lösningen på problemet är inte alltid samma sak. Som Kruchten menar: *"Det tillfälle då användarna vet exakt vad de vill ha är inte två år innan systemet är klart, utan snarare några veckor eller månader efter att systemet har levererats, då de passerat första inlärningsfasen."* (Kruchten, 2002, s. 56) Detta har enligt Kruchten blivit kallat IKIWISI-effekten (I'll Know It When I See It), vilket är ett uttryck även använt av Boehm (1988).

Det är dessutom så att det inte bara är förändringar i kraven som gör att de är näst intill omöjliga att specificera i inledningsskedet av projektet. Gulliksen & Göransson (2002) menar till exempel att det inte går att skapa en statisk kravspecifikation innan utvecklingen har påbörjats bland annat på grund av att specifikation, design och konstruktion inte kan ses som separata arbetssteg över tiden. Man är ofta inte tillräckligt insatt i problemområdet innan man kommit längre fram i utvecklingsprocessen.

3.3.2. Förändringar i förutsättningar

Det är dock inte bara förändringar i krav som påverkar projektets resultat. Många andra faktorer finns runt projektet som man ofta inte reflekterar särskilt mycket över i projektet. Några av de faktorer som bland annat Kruchten (2002) talar om här är till exempel förändringar i:

- Teknik

Om projektet är långvarigt och kanske sträcker sig över flera år (i dagsläget går utvecklingen så snabbt att det dock kan räcka med några månader) så finns risken att de tekniska plattformar man valt att använda sig av slutar tillverkas eller bedöms undermåliga eller för gamla mot slutet av projektet. Det kan vara gamla programspråk, gamla databasservrar eller gamla operativsystem.

- Marknaden

Andra företag som finns på marknaden kan eventuellt vara på väg att utveckla liknande produkter eller andra produkter som löser samma problem. Som beställande organisation kanske man gör bättre i att köpa in ett färdigt standardsystem som lanseras mitt under projektets gång istället för att låta projektet löpa vidare.

- Tillgängliga resurser (tid, pengar och personal)

Systemutvecklingsprojekten har så gott som alltid någon form av budget såväl för hur mycket pengar det beräknas kosta, hur lång tid det bör ta och hur mycket arbetskraft som krävs. Detta är tyvärr inget som är hugget i sten utan är något som ofta hålls igen från beställarens sida. Det måste kanske vara klart till ett visst datum eller maximalt kosta en viss summa.

- Tillgång till ledningens stöd

Hur bra stöd man har från ledningen hos det beställande företaget är något som kan vara direkt avgörande för hur väl projektet lyckas eller inte. Andersen et al. (1994) menar att just detta är en av de viktigaste punkterna att ta hänsyn till vid projektets start. Detta är också föremål för förändring, då till exempel nyckelpersoner i en ledningsgrupp kan ersättas av andra, eller få minskad makt till exempel på grund av order från koncernledning eller dylikt.

Detta är endast några exempel på alla de faktorer som kan finnas i ett projekts omvärld och som faktiskt kan påverka resultatet negativt, men självklart finns det många fler. Just att möta alla typer av interna och externa förändringar är ett centralt problem med de sekventiella modellerna vilket också Sommerville (1996) och Beck (2001) talar om.

3.3.3. Fokus på dokumentation

Såväl Kruchten (2002) som Gulliksen & Göransson (2002) menar att de sekventiella modellerna i hög grad fokuserar på att producera dokumentation. Kruchten (2002) menar till exempel att alla faser utom den sista har som primärt mål att producera någon form av dokumentation. Dessutom ligger också fokus på att färdigställa och frysa dessa dokument för att använda detta som indata i nästföljande fas.

Gulliksen & Göransson (2002) menar dessutom att den dokumentation som framställs ofta är alldeles för tekniskt orienterad, formell och alldeles för komplex, till och med i den höga grad att många utvecklare har svårt att förstå dem. Man kan då bara börja reflektera över hur komplex den då är för den beställande organisationen, som dessutom ska godkänna delar av dokumentationen, framför allt de inledande specifikationerna.

Att skriva denna komplexa dokumentation leder också till att man i systemutvecklingsprojekten ofta gör mycket onödigt arbete. Boehm (1988) skriver till exempel *"Document-driven standards have pushed many projects to write elaborate specifications of poorly understood [problems] followed by the design and development of large quantities of unusable code"*. (Boehm, 1988, s.63) Det kan alltså inträffa om man har dålig kunskap om problemområdet i början av utvecklingsprojektet, vilket inte är ovanligt i stora systemutvecklingsprojekt. Vidare kan det också inträffa om krav eller förutsättningar förändras och gör vissa delar onödiga att utveckla. Då har man redan utfört analyser och designat dessa delar och skrivit tunga specifikationer för dem, vilket är ett arbete som inte kan göras o gjort (Beck, 2001).

3.3.4. Tekniskt fokus och användardistansering

Gulliksen & Göransson menar som vi beskrev i föregående avsnitt att dokumentationen ofta är tekniskt komplex. Detta hänger ihop med att de också anser att hela fokus i systemutvecklingsprojekt med sekventiellt arbetssätt är allt för tekniskt inriktat. De menar att hela upplägget med inställningen att man på förhand kan sätta upp en lösning på ett problem som struktureras upp och definieras på ett logiskt sätt är felaktig, åtminstone i systemutveckling som vi väljer att se begreppet. För ren mjukvaruutveckling skulle dock denna modell eventuellt kunna vara tillräcklig. Författarna menar att detta skapar en ingenjörprocess där man delar upp problemet i delar som ska lösas en efter en för att nå en fram till det förutsägbara målet. (Gulliksen och Göransson, 2002)

Detta gör också att man i utvecklingsarbetet tenderar att förlora kontakten med slutanvändarna, vilket ofta leder till att systemet blir mindre användarvänligt än vad det kunnat bli med en tydlig användarcentrering (Gulliksen & Göransson, 2002). Gulliksen & Göransson (ibid.) menar att man i detta tekniska fokuserande ofta tenderar att se användare som inte riktigt vet vad de vill ha som ett osäkerhetsmoment som skapar en risk för projektets resultat och att man således bör undvika att ta in dem i utvecklingsprocessen. Tyvärr är sanningen ofta det motsatta; systemet kan eventuellt bli jobbigare att ta fram men blir då istället ett system som möter rätt problem på ett korrekt och tillfredsställande sätt. Zeffane et al. (1998) menar att denna användarmedverkan tidigare varit omdebatterad och att vissa forskare kommit fram till motsatta resultat, men att man numer generellt sett är överens om att användarmedverkan har en positiv inverkan på resultatet. Författarna menar till exempel att:

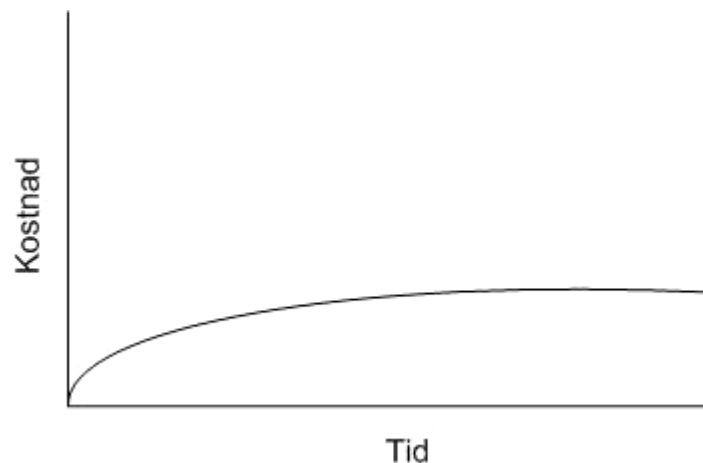
...if information systems departments are to contribute to corporate strategy, they must be able to work hand in hand with **departments** and work groups toward strategic goals. To enable that cooperation, IS needs to bridge the considerable gap that often exists between itself and the functional users (Zeffane et al, 1998 s.115)

Detta uppnår man lättare genom en hög grad av användarmedverkan. Författarna menar vidare också att användarmedverkan också leder till bättre problemlösning och konflikthantering, vilket påverkar projektets resultat markant. (Zeffane et al, 1998) Andra författare, däribland Cavaye (1995) talar om att användarmedverkan är viktigt då det är av högsta relevans att man får med alla perspektiv på de krav som finns på det framtida systemet. Om denna nackdel kan hittas i vår undersökning kan det således vara mycket problematiskt, då man hamnar i en nästan paradoxartad situation. Ett sekventiellt arbetssätt förutsätter en ingående kravspecifikation vilket i sin tur förutsätter en hög grad av användarmedverkan, vilket i så fall skulle motverkas av användandet av ett sekventiellt arbetssätt.

3.4. Finns samma nackdelar hos de evolutionära modellerna?

Hur kan vi då säga att de nackdelar vi tagit upp här faktiskt är nackdelar med de sekventiella modellerna och endast dessa? Risker finns ju att vissa av de nackdelar vi tagit upp är nackdelar med systemutveckling i stort och inte nackdelar enbart för de sekventiella modellerna. Detta vill vi dock visa på med hjälp av att titta på hur de evolutionära modellerna handskas med dessa problem. Detta ska dock inte missuppfattas. Självklart har också de evolutionära modellerna nackdelar och brister som kan påverka resultatet negativt, men det är inte intressant för denna studie att ta upp dessa. Vi är också medvetna om att de författare och forskare som vi tittat på i detta kapitel ofta stödjer en systemutvecklingsprocess som är baserad på de evolutionära modellerna. Detta gör att den kritik som uppkommer hos dessa författare kanske är onödigt hård, men detta är just en av de anledningar till varför vi vill ta upp detta för närmare granskning.

Börjar vi med de sekventiella modellernas inneboende problem med att hantera förändringar, såväl i krav som i kringliggande förutsättningar, så är den klassiska bilden att kostnaden att förändra systemet ökar exponentiellt med tiden. Beck (2001) menar att detta skulle kunna vara tvärt om; kostnaden skulle kunna plana ut mer och mer allt eftersom tiden går. Man får en graf som liknar:



Figur 3-6: Kostnad för förändring i evolutionära modeller

Detta skulle istället ge beställaren möjlighet att istället vänta med stora beslut till sent i systemets utvecklingsprocess. Just detta åstadkoms genom att man utför en mycket kort och övergripande planering och bestämmer egentligen vilka delar som är viktigast att inleda utvecklingen med och vilka som kan vänta till kommande iterationer, då man bygger på de första delarna. På så sätt utför man bara analys, design och konstruktion på de delar som faktiskt är relevanta att utveckla. (Beck, 2001)

Här använder man sig dessutom av en approach som är mer resultatdriven än dokumentdriven. Varje del utvecklas snabbt utan särskilt mycket tung dokumentation, ofta med hjälp av prototyper, vilket gör att problemet med tung dokumentation försvinner. Delarna utvecklas vidare i nära samarbete med kunden för att hela tiden ha en hög grad av användarmedverkan. Detta gör också att det tekniska ingenjörsvetenskapliga fokus vi har i de sekventiella modellerna i stort sett försvinner helt. (Beck, 2001)

3.5. Sammanfattning

Med utgångspunkt i den valda litteraturen har vi skapat vad man skulle kunna se som ett teoretiskt ramverk som den fortsatta studien kommer att bygga på. Nackdelarna som vi har kunnat hitta i litteraturen är de som vi kommer att bygga vidare på för att på så vis ha en stadig grund att stå på i den empiriska undersökningen.

Vissa av de nackdelar vi tagit upp är betydligt allvarligare än andra. Till exempel kan man tycka att nackdelen fokus på dokumentation egentligen är ganska intetsägande. Vi vill dock mena att så inte är fallet. Ett allt för stort fokus på dokumentation kan kanske göra att man lätt tappar bort för kunden mer relevanta delar av projektet som till exempel det faktiska designarbetet eller själva konstruktionen av det. Detta kan då leda till att kunden ser snett på projektet eftersom man bara sitter i möten och skriver en massa planer och specifikationer istället för att göra något ”verkligt” arbete. Att dokumentation är nödvändigt är dock inget vi vill säga emot, men att man kan lägga olika mycket fokus på dessa delar, och de helt iterativa modellerna väljer att lägga väldigt lite resurser på detta.

Andra nackdelar kan vara betydligt värre om det visar sig att de är tydliga. Man kan här återknyta till det resonemang vi tidigare fört; om det är så att man tenderar att distansera sig från användarna så hamnar man i ett slags moment 22, där det sekventiella arbetssättets förutsättningar – tydliga specifikationer – förutsätter hög grad av användarmedverkan.

Dessutom hör grupperna av nackdelar bitvis ihop. En bristande användarmedverkan kan till exempel leda till att kunden inte lyckats förmedla sina önskemål på ett bra sätt. Det i sin tur kan leda till att projektets systemkrav förändras varefter kunden inser att det som utvecklas inte är det som man vill ha. Dessa förändrade krav leder då till fler nackdelar och risk för problem.

Det ska tilläggas att vissa av de nackdelar vi tagit upp i kapitlet, ibland har kommit från förespråkare av mer iterativa processer. Detta kan innebära att åsikterna kan vara ganska partiska, men det leder inte till att nackdelarna blir mindre intressanta att undersöka; det kan ju vara riktigt att de faktiskt är problem för de sekventiella modellerna.

4. Fallstudien

I denna del av uppsatsen presenterar vi vår fallstudie och sammanställer dess resultat inför analysen i nästa kapitel.

4.1. Struktur

I vårt arbete med att undersöka våra fall och försöka sammanställa de data som vi samlat in studerade vi alltså tre olika fall. Nedan kommer dessa fall att ett i taget först presenteras med en beskrivning av projektet, sedan går datainsamlingen igenom varpå vi slutligen följer upp detta med en sammanställning av de data som vi samlat in i fallet. Observera att data här endast presenteras; vi gör ingen värdering eller tolkning i denna del, förutom det som automatiskt följer med en sammanställning. Intervjuanteckningarna, som för övrigt återfinns i bilagorna, är inte sammanställda utan består enbart av de anteckningar vi tagit på plats tillsammans med respondenterna. I dessa anteckningar kan man återfinna ett flertal värdeladdade påståenden, som således endast är en redogörelse för vad respondenten sagt under intervjun. Det har alltså inte vävts in några egna analyser eller slutsatser, varken i intervjuanteckningarna eller i detta kapitel.

De tre fallen ligger sedan till grund för analysen i nästa kapitel där vi bryter ned data, gör en fallövergripande sammanställning, analyserar de data vi har och slutligen i kapitel 6, tillsammans med det vi kom fram till i kapitel tre, utgör grunden ifrån vilken vi drar våra slutsatser.

4.2. Fall ett

4.2.1. *Fall ett – beskrivning*

Leverantör i projektet var ett mindre utvecklingsföretag som hade en av sina anställda deltidsanställd som IT-ansvarig hos kunden. Kund var ett medelstort konsultföretag inom läkemedelssektorn.

Utvecklingsprojektet inleddes med att kunden insåg att den lilla applikation, för övrigt skriven i Access, i vilken de rapporterade in sina konsulter upparbetade tid snabbt blev begränsad varefter företaget utvecklades och expanderade. Detta var främst på grund av ändrade behov och på grund av att en undermålig specifikation låg som grund. Istället föddes planer på att koppla samman tidrapportering med fakturering. Det skulle skapa ett mer fullständigt affärsstödssystem med integrerad funktionalitet som kunden ansåg sig ha behov av.

Själva utvecklingsarbetet inleddes år 2000 med en budget på 400 000 kr. I projektgruppen ingick sex personer. En beställning genomfördes i projektets inledning, i huvudsak av projektledaren som samtidigt var anställd på kundföretaget. Dock så stämde alla idéer av med företagsledningen innan projektet inleddes. Ett grundläggande krav på systemet var att det skulle kunna användas oberoende av geografisk plats med tanke på företagets expensionsplaner. Därför utvecklade man en webbaserad applikation.

4.2.2. *Fall ett - datainsamling*

Respondenter från fallet var dels projektledaren Kalle, och dels en av programmerarna, David.

Kalle är en man som nu är 28 år gammal. Han arbetar fortfarande kvar i kundföretaget på deltid och tycker om sin delade arbetssituation. David är också en man i samma ålder som Kalle, men då han endast var anställd i projektet för den tid som det löpte har han nu ett annat jobb som utvecklare på ett mindre IT-företag.

Intervjuerna genomfördes vid olika tillfällen om cirka en till två timmar med de båda respondenterna. Vid båda tillfällena togs det anteckningar på en laptop. Båda var väldigt mån om att få behålla anonymitet då det eventuellt skulle komma fram för respondenterna ofördelaktiga resultat ur intervjun.

Vidare så fick vi, förutom intervjuer, tillgång till den dokumentation som man hade producerat under utvecklingsarbetet. Detta, i samverkan med att vi även fick tillgång till att titta på själva systemet, gav oss en god grund för insamlande av data.

4.2.3. *Fall ett - utfall*

När utvecklingsprojektet inleddes beslöt man sig i projektgruppen för att använda sig av den enda tekniska plattform som man kände till: ASP och IIS. I projektgruppen ingick en person som agerade projektledare och som därmed var ansvarig för utvecklingen. ASP och SQL utvecklades i huvudsak av tre av de övriga inkluderade och slutligen så hade man två personer som ansvarade för Javaklienten. Det skulle byggas två gränssnitt; webbgränssnittet som primärt var det tidrapport- och faktureringsgränssnitt som användarna skulle jobba i och sedan även ett administratörsgränssnitt (Javaklienten).

Enligt respondenterna arbetade man inte direkt efter någon specifik utvecklingsprocess, utan man använde istället en modell som delade upp arbetet i fyra faser (analys, design, konstruktion och test/implementering) som grund för arbetet. Detta kändes enligt respondenterna som ett naturligt och okomplicerat sätt att lägga upp arbetet.

Enligt Kalle så avsattes små resurser (inte adekvata) inledningsvis till analys och design. Dessutom användes resurserna på ett felaktigt sätt. Till exempel förväntades systemet vara modellerat på en vecka, vilket skapade stor tidspress. Detta berodde på att systemets storlek underskattades och man trodde även att det var enbart databasen som egentligen var i behov av design vilket innebar att nästan all inledande analys och design lades till att skapa just databasen. Detta ledde bland annat till att funktionaliteten i gränssnittet inte designades med helt oläsbar kod som följd. Detta insåg inte Kalle eftersom han inte deltog i själva kodarbetet. Det var först när man försökte göra anpassningar som

Kalle insåg att koden var omöjlig att underhålla och omöjlig att utveckla vidare. Detta ledde senare till att kodbasen var tvungen att skrivas om i ett uppföljningsprojekt.

Resurser till återkoppling saknades helt i projektet just eftersom man arbetade efter en systemutvecklingsmodell som innebar att man utförde arbetet med analys tidigt i projektet för att skapa en kravspecifikation. Detta skulle sedan inte förändras utan om man hade gjort ett fullgott arbete med kravspecifikationen så skulle detta kunna skapa en fullgod grund för resten av arbetet osv. Enligt respondenterna så berodde denna inställning, och valet av systemutvecklingsmodell, framförallt på okunskap. David menar att: *”Hela projektet var som en läroprocess. Vi lärde oss otroligt mycket på vägen.”*

Kalle medger dock att man även kan se det som en kombination av brist på resurser som tid, pengar och motivation. Det var lättare att anamma en sekventiell modell eftersom det verkade lättare att få en översikt och dessutom verkade ständiga iterationer vara kostsamma.

Under arbetet var Kalle marginellt medveten om att modellen innebar nackdelar, det är en lärdom som han känner att han kan dra ur processen samt ur resultatet. Om han hade haft bättre koll så hade han sett till att avhjälpa situationen. De andra deltagarna hade ännu mindre koll på detta, vilket bekräftas av David. Detta tror Kalle främst beror på att ingen i projektet hade någon kunskap eller akademisk grund med sig i bagaget.

Systemutvecklingsmodellen var inte alls av intresse för kunden. De tyckte enligt Kalle att de hade ett tydligt behov som man utan problem kunde dokumentera i de inledande faserna av projektarbetet. Man såg inte att det skulle kunna uppkomma några större framtida behov och ville inte lägga resurser på att till exempel återkoppla till analys och design. Detta till trots att Kalle även var beställare och hade inflytande på detta. Detta berodde dels på en oförmåga hos Kalle att förmedla vikten av en förmåga att hantera förändring i krav och förutsättningar, dels på beställarens dubbla roll (enligt Kalle så tappade han i trovärdighet när han både var anställd samt arbetade som extern konsult) samt även av okunskap hos kunden (Kalle: *”Man hade bristande insikt i systemets omfattning och i betydelsen av systemet för sin verksamhet.”*) och slutligen även ekonomiska faktorer.

Det sätt man valt att lägga upp aktiviteterna i projektet på, alltså projektets faser, bidrog enligt Kalle inte till att distansera utvecklingsarbetet från systemets framtida användare. Detta eftersom de aktiviteter som användarna var involverade i och för sig lades tidigt i projektet men då löpande genomgångar med användarna genomfördes ända fram slutttestningen av systemet fick användarna ändå en känsla av delaktighet. Detta trots att de enligt David i själva verket hade lite inflytande över design- och konstruktionsfaserna.

Projektdeltagarna överlag såg enligt respondenterna ett evolutionärt tänkande som väldigt ”jobbigt” och komplicerat. De tyckte det var mycket bättre att sätta upp en specifikation och en systemdesign för att sedan sätta sig och koda. När man kodade vill man helst inte bli avbruten med andra aktiviteter utan istället var det viktigt att man såg snabba resultat i form av funktionalitet. De flesta såg dessutom enligt David inte processen som en läroprocess, man ville inte riskera kritik genom att till exempel återkoppla till analys och design. Man ville inte behöva se att man inte gjort rätt, att man inte skulle ha konstruerat systemet på ett fullgott sätt eftersom kraven förändrats eller liknande. Det skulle ju innebära att man skulle behöva ta konsekvenserna av vad man skulle upptäcka i och med återkopplingen. David: *”Det sågs som ett riskmoment.”* Det kunde även enligt Kalle ses som beroende på ren lathet. Man orkade inte åtgärda fel som kunden inte hade märkt. *”Om de inte märker att det är finns fel så finns det inga fel”* var enligt Kalle en stark uppfattning hos de andra deltagarna.

Det visade sig efter ett par månaders arbete att beställningen och planeringen var orealistiska på grund av att företagsledningen inte hade koll på innebörden och konsekvenserna av att ha projektredovisning som bas för systemet. Man hade tydligt i inledningsskedet angivit att planeringen skulle bygga på att all verksamhet i sin tur byggde på projekt. Fakturering skulle ske på projektbasis. När man demonstrerade en av demoversionerna av systemet för kundföretagets VD så insåg han dock att projekt inte alls borde ligga till grund för en faktura utan till exempel skulle flera projekt kunna finnas på samma faktura. Fakturan skulle alltså vara kundbaserad. Detta innebar att man enligt Kalle satt, på grund av en kommunikationsmiss, med ett system som byggde på en felaktig grund. Detta innebar att man var tvungen att göra om en del av arbetet, vilket upplevdes som mycket irriterande av David. Detta innebar dessutom problem för projektet att hålla tidsramarna när nu produkten var tvungen att byggas om, vissa delar i stort sett från grunden.

Ett annat stort praktiskt problem som man stötte på i utvecklingsarbetet var att det visade sig att det fanns ett behov av fakturor på flera sidor, vilket inte Internet Explorer (som var vad de först använde för att rendera fakturor) klarade. Man var då istället tvungen att använda sig av PDFer för att visa fakturor. Mycket var tvunget att skrivas om när det var dags att porta fakturorna till PDF då man inte hade använt sig av något objektorienterat tänkande och ingen egentlig återanvändning av kod eller andra tekniker. Uppgiftens förändring i och med tekniska problem tvingade alltså fram ytterligare en omarbetning av delar av produkten.

Vad man enligt Kalle kan konstatera från slutresultatet av första delen av projektet är att då man inte hade någon tanke på iterationer eller återkoppling var man tvungen att skriva om stora delar av kodbasen eftersom den inte gick att använda för någonting annat än precis det den var specificerad för, alltså det man hade definierat i de tidiga faserna i projektet. Hade man haft en bättre hantering av analys och design och återkopplat till detta arbete regelbundet hade man troligen kunnat se riskerna och bristerna snabbare. Kanske hade man då insett vikten av ett generellare system för just dessa förutsättningar. David: ”[Det är] *definitivt så att bristen på återkoppling har påverkat projektets resultat negativt.*”

Vad gäller slutresultatet så anser respondenterna att kunden på det hela taget var nöjd med funktioner och användbarhet. Dock så var de inte riktigt lika nöjda med att projektet kraftigt hade överskridit sin tidsplan. Till saken hör också att respondenterna och resten av utvecklingsgruppen lade ned mer tid än vad de fick betalt för eftersom den inledande uppskattningen av kostnaderna inte höll.

Vidare kan man enligt båda respondenterna klart se att synen på sekventiellt arbete förändrats efter utvecklingsprojektets avslut, speciellt Kalle som varit så kraftigt involverad i arbetet med analys och design. Återkoppling till analys och design blev senare ett viktigt moment vid vidareutveckling av systemet.

4.3. Fall två

4.3.1. Fall två – beskrivning

I detta utvecklingsprojekt var leverantören ett medelstort utvecklingsföretag med webbaserade affärslösningar som sin specialitet. Företaget hade vid tidpunkten cirka 12 anställda. Kunden var ett ganska stort företag som tillverkar specialgjorda ventilationssystem.

Tankarna kring projektet uppkom 2001 i och med att Rikard träffade ansvariga hos kunden på en privat tillställning. Detta ledde till att ett professionellt samarbete inleddes under 2002 då kunden uttryckte önskemål kring hjälp med ett systemutvecklingsprojekt som de var på väg att inleda. Det gällde utveckling av en applikation som kunde hjälpa dem att utföra tekniska beräkningar på ett bra sätt. Tidigare hade man använt sig av Excel, men det var inte längre tillräckligt. Systemet skulle även vara tillgängligt för deras återförsäljare, som återfinns i ett flertal länder, att använda. Alltså var en webbaserad lösning att föredra.

Projektet hade ingen formell projektledare (Rikard blev dock något av en informell projektledare under resans gång då det var denne som hade kontakt med kunden och som gjorde de avstämningar som man gjorde mellan kund och utvecklare) utan endast en ansvarig från kund. Istället bestod själva projektgruppen av fem personer på samma nivå. Två av dessa var respondenterna. Alla kunde ses som fem yngre personer utan utförligare formell utbildning som gjorde detta lite för att det var kul och spännande och lite för att tjäna pengar. För respondenterna var inte pengarna i fokus. Beställningen av projektet genomfördes på möten mellan projektgruppen och personer från kund. En grundtanke var att systemet skulle vara generiskt och fungera till även annat är det som var primärt. Detta var en linje som framförallt projektgruppen drev. En offert skrevs och lämnades över vilken innehöll ett fast pris.

4.3.2. Fall två – datainsamling

Respondenter från fallet var dels den informella projektledaren, Rikard, och dels en av de andra utvecklarna, Caroline.

Rikard är en man som nu är 26 år gammal. Han arbetar fortfarande med systemutveckling, fast nu ofta som formell projektledare. Caroline är nu 27 och arbetar numer som databasutvecklare på heltid.

Intervjuerna genomfördes vid olika tillfällen om cirka en till två timmar med de båda respondenterna. Vid båda tillfällena tog vi anteckningar på en laptop, precis som i fall ett ovan. Båda var mån om att få behålla sin anonymitet.

Vidare så fick vi, förutom intervjuer, precis som i fall ett ovan tillgång till den dokumentation som man hade producerat under utvecklingsarbetet. Även här fick vi tillgång till att titta på själva systemet.

4.3.3. Fall två – utfall

Enligt Rikard var detta ett väldigt problematiskt projekt, av olika anledningar. Arbetet inleddes med att man lade upp en planering som byggde på någon form av sekventiell metod med på varandra följande delar. Det första man gjorde i projektet var att ta fram en kravspecifikation i någon form av analysfas. Detta följdes sedan upp genom att man designade de delar som krävde detta, till exempel databasen, för att sedan börja skapa funktionalitet. Slutligen så testades alltihop och levererades till kund.

Enligt Rikard slutade dock analysen i en tämligen svag kravspecifikation, kanske främst beroende på att offerten och beställningen var ganska vag och man endast trodde man hade koll på vad det var man ville göra, vilket kom att förändras.

Redan tidigt i projektet ändrades nämligen förutsättningarna. Nya saker som ”*skulle vara bra om det var med*” kom upp och dessa involverades löpande i projektet utan någon återkoppling till planering, analys eller design. Det hade enligt Rikard varit bättre om dessa saker hade samlats upp och sammanställts till någonting man kunde komma tillbaka till i en framtida vidareutveckling av produkten så att man kunde involvera den nya funktionaliteten i projektet även i dess analys- och designrelaterade delar och inte bara i konstruktionen. Så gjorde man hur som helst inte.

Det innebar inte så stora problem under den första fasen, men senare i projektet när man hade en lösning relativt färdigkonstruerad och man hade det man trodde man skulle testa för att sedan leverera uppstod problem. Här trodde Caroline att hennes involvering i projektet var avslutat. ”*Då hade vi skapat det som jag var involverad för att ta hand om.*” Det visade sig dock att kunden inte alls delade uppfattningen om att det var så systemet skulle se ut när det levererades. Man hade ju nya önskemål om funktionalitet som många uppkom först när man kom i kontakt med en nästan färdigställd version av systemet.

Under projektets gång var respondenterna ganska medvetna om bristerna i projektarbetet. Dock så jobbade man enligt Rikard enligt approachen ”*Vi lägger till detta så fixar vi det också*”. Det var väldigt viktigt för projektdeltagarna, kanske framför allt för Rikard som hade mest kontakt med kunden, att kunden blev nöjd med resultatet. Bristen på iteration upplevdes dock inte som ett stort problem av respondenterna under projektets gång. Detta tror Caroline berodde på att man såg det som väldigt positivt att nå resultat, främst då i form av funktionalitet, vilken man ju gjorde. Problemet var snarare att resultaten inte kunde hållas ihop eftersom kraven förändrades hela tiden. Eventuella skillnader mellan de olika deltagarna berodde enligt Caroline på brist på insikt i projektet, personlighet samt arbetsuppgifter och ansvar.

Respondenterna anser nu i efterhand att ett fasövergripande arbete med analys, design och konstruktion är väldigt viktigt. Problemet i projektet var dock att respondenterna inte hade dessa kunskaper då och dessutom var de enligt Caroline för snälla. ”*Vi sade inte ifrån fast vi insåg att alla nya förändringar skulle ge ett sämre resultat om de lades till löpande under arbetets gång.*” Nya funktioner inkluderades alltså på löpande band utan att någon sade emot och det tar Rikard på sitt ansvar eftersom han var den som skötte kontakten med kund och som inte berättade för dem att de inte bara kunde ändra målen för projektet utan att till exempel förändra andra förutsättningar och analysera de nya behoven. Det står enligt respondenterna klart att avsaknaden av återkoppling till både projektplanering och till analys/design kraftigt påverkat projektets resultat negativt, framför allt vad gäller systemets användbarhet och möjlighet till vidareutveckling. Rikards uppfattning om nackdelarna med sekventiella model-

ler har förändrats mycket, eller snarare kommit till, under projektets gång. ”*Man är inte dummare än att man lär sig.*”

Rikard var involverad i hela projektarbetet och var nog den starkast drivande kraften. Han kände att det var lite hans ansvar att det skulle bli bra eftersom det ju var han som höll den mesta av kontakten med kunden. Alla projektdeltagare var dock involverade, och det var även representant från kunden i viss mån. Kunden var dock endast involverad i den inledande planeringen, alltså i den första av de sekventiella faserna.

Vad som är intressant att nämna här är att det fanns bra med resurser för projektarbetet, framför allt eftersom deltagarna använde även sin fritid till projektet. Det är svårt att avgöra vart resurserna i form av ekonomiska medel gick då man inte tog betalt för all den tid man lade ned i projektet. Det var ju lite för att det var roligt och spännande. Tidplanen var på det hela taget helt ok, Caroline menar att ”*Vi hade rätt gott om tid*”. Det fanns med andra ord gott om tid till utvecklingsarbetet. Detta gällde naturligtvis även för eventuell evolutionär iterering mellan aktiviteter som man dock ändå inte begagnade sig av. Enligt Rikard var ”... *resurser inte begränsade*”.

Uppgiftens föränderliga karaktär gjorde att det fanns ett stort behov av återkoppling, speciellt eftersom kundens önskemål under projektets gång har förändrats nästan lite från dag till dag under perioder. Uppgiften upplevdes av respondenterna som väldigt komplex. Rikard menar att det sekventiella upplägget av arbetet påverkade resultatet negativt, främst med tanke på hur missnöjd kunden var med vissa delar av funktionaliteten samt med avseende på tidsplanen som man överskred. Kostnaderna hölls inom ramarna främst eftersom man inte fakturerade för all arbetstid.

Vad gäller kundens reaktion på den slutliga produkten så verkar den på det hela taget dock vara bra. Respondenterna upplever att kunden är nöjd med användbarheten i systemet och för att systemet, trots vissa brister i funktionaliteten, löser de problem som gjorde att arbetet inleddes.

Under de senare delarna av projektet så försökte man ta upp någon form av återkoppling till tidigare delar även om det fortfarande var bristande. Man ville försöka skapa sig en ny bild av vad det var man skapade. Det visade sig dock mycket svårt att förändra arbetsmodellen under projektets gång. På det hela taget har respondenterna fått en förändrad inställning till sekventiellt arbete. Det känns fortfarande som ett naturligt sätt att lägga upp arbetet på men man är numer medveten om vikten att löpande gå tillbaka till tidigare faser för att uppdatera det arbetet.

Vad respondenterna upplever var de största nackdelarna med det sekventiella upplägget i projektet var att resursbehovet påverkas negativt av att man inte var rustad att möta förändrade krav från kund eller förändrade förutsättningar. Istället fick man investera mer resurser i form av arbetstid för att det skulle gå ihop i slutändan. Dessutom anser Caroline att det vore mycket bättre om kunden involverades mer under hela arbetet eftersom det då skulle ge möjlighet till mer regelbundna avstämningar vilket förhoppningsvis skulle kunna leda till att man kunde undvika reaktioner av den typ som kom från kund när de såg systemet strax innan leverans. Det var ju inte alls vad de ansåg att man hade kommit överens om, kanske mest på grund av att projektdeltagarna hade låtit kunden modifiera specifikationen muntligen under arbetets gång.

4.4. Fall tre

4.4.1. Fall tre - Beskrivning

I augusti 2003 startade den svenska delen av ett stort marknadsföringsföretag, som finns representerat i stora delar av världen, upp ett dotterbolag som först skulle utveckla och därefter underhålla ett webbaserat spelcommunity. Grundidén med detta utvecklingsprojekt var att skapa en gemensam plattform för alla spelare, oavsett om det var datorspel, rollspel eller kortspel man var intresserad av även om primärt fokus låg på datorspelare. Här skulle medlemmarna kunna chatta, diskutera och kanske framför allt spela datorspel. Deadline för projektet var att communityt skulle vara i drift vid årsskiftet 2003/2004, med en budget på 1,5 miljoner.

Från början togs, utöver projektledaren, fyra anställda in som skulle ansvara för utvecklingen av communityt.

De spel som medlemmarna skulle kunna spela var framför allt mindre flash- och shockwavespel, som alla hade utvecklats särskilt för detta spelcommunity. Dessa spel skulle kunna spelas i turneringar eller bara för skojs skull. Här kommer vi dessutom in på företagets affärsidé. För att tjäna pengar lät företaget medlemmarna betala för att få spela spelen i communityt. Valutan var i form av virtuella mynt som man fick köpa för verkliga pengar. Turneringar skulle kosta mer att ställa upp i, men skulle också ge möjlighet att vinna en större summa virtuella mynt.

Detta företag var relativt självstyrande utan större inblandning från det ägande företaget, vilket gjorde att det inte ställdes särskilt mycket krav på hur systemet skulle fungera. Detta ägarföretag kan dock ändå ses som projektets primära kund, även om inte slutanvändarna finns inom detta företag.

4.4.2. Fall tre - Datainsamling

Den främsta källan vi använt oss av i undersökningen av detta fall var intervjuer med två personer i den ursprungliga utvecklingsgruppen. Den första respondenten, Christian, var projektets projektledare. Han är idag 36 år och har en ledande position inom det tidigare ägarföretaget, som han påbörjade strax efter det att utvecklingsverksamheten och arbetet i projektet lades ned.

Den andra respondenten, Nils, är idag 26 år och inte längre anställd varken i projektet eller i något företag kring detta. Han arbetar istället som programmerare på ett annat medelstort IT-företag.

Förutom dessa intervjuer har vi fått ta del av såväl delar av den dokumentation som producerats i projektet, som mycket av den dokumentation som medlemmarna genererat i till exempel foruminlägg och dylikt.

4.4.3. Fall tre - Utfall

Projektet inleddes under hösten 2003 med att man först och främst gjorde analys av communityts önskade struktur. Man skulle till exempel bestämma vilka delar som skulle inkluderas i det framtida

systemet och hur de skulle fungera. Dessutom var det av vikt att bygga upp strukturer som till exempel vilka olika typer av medlemmar som skulle finnas. Det skulle till exempel kosta olika mycket att delta i turneringar för olika typer av medlemmar. Vidare gjorde man även en teknisk analys och beslutade vilka verktyg och plattformar man skulle använda i projektet.

Kravspecifikationen som framställdes lade som sagt grunden för vilka delar som skulle utvecklas. En av de delar som systemet skulle innehålla, var ett generellt turneringssystem som skulle kunna användas till alla typer av spel, såväl de enkla flashspel man tänkte lägga i communityt som mer avancerade datorspel som Counter Strike eller Quake 3. För att dessa skulle fungera så ville man även sätta upp egna spelservrar som skulle interagera med turneringssystemet för att rapportera resultat och dela ut priser. Med hjälp av dessa servrar som dragplåster skulle framför allt många datorspelare lockas till communityt.

Förutom dessa delar av systemet skulle även ”vanliga” community-funktioner skapas. Man ville ha möjlighet att skapa sin egen profil, ha en Gästbok som andra skulle kunna skriva i, möjlighet att skicka mail till andra medlemmar och möjligheter för chat i realtid.

Man arbetade inte direkt efter någon specifik systemutvecklingsprocess, utan grundtanken var framför allt att man skulle arbeta med ett klassiskt upplägg, med de klassiska vattenfallsfaserna eftersom detta kändes som ett naturligt sätt att arbeta. Detta var inget som man reflekterade särskilt mycket över, då de flesta som arbetade i projektet var programmerare eller andra med framför allt tekniskt fokus. Christian var dock, trots att han var projektledare, inte särskilt väl insatt i systemutvecklingsmetodik utan överlät mycket av själva arbetet med detta åt de övriga projektdeltagarna. Christian säger själv att: *”Jag hade framför allt en säljande funktion; jag sålde in konceptet dels till ägarna och dels till projektdeltagarna själva, sen fixade de allting själva.”* Nils menar istället att: *”Det var skönt att kunna fatta mycket av besluten själv, men det märktes ju hur dåligt det var när allt blev mer komplext.”*

Då hela systemet skulle designas valde man att designa allt med hjälp av verktyg från den objektorienterade världen. UML användes för att skapa en systemspecifikation för själva koddelen som skulle vara lätt att förstå och lätt att underhålla och modifiera. För databasnivån användes inga direkta modelleringsmetoder utan man skapade den lite grann som man kände passade bra. *”Tyvärr hade vi som designade systemet alldeles för dålig kunskap om objektorienterat tänkande. Vi skapade en alldeles för teknisk komplex design som inte alls överensstämde med det ändå ganska enkla communityt. Till slut blev det helt ohållbart att göra förändringar i systemet.”* (Nils)

För designen av användargränssnittet gick man till en extern reklambyrå som fick komma med en grafisk profil. Profilen inkluderade även ett förslag på layout för communityt, och en logga, men huruvida funktionaliteten som lades in i den grafiska profilen var användarvänlig eller inte studerades aldrig.

Dessa två första faser gick igenom relativt snabbt, men genererade ändå låsta krav- och systemspecifikationer. Då själva konstruktionen påbörjades stötte man på ett antal stora problem. Dels var *”systemet [...] så komplext designat att det tog fruktansvärt lång tid att följa systemspecifikationen, även om vi då ansåg att det var värt det då det skulle vara skitenkelt att underhålla och vidareutveckla”*. Man stötte även på ett antal stora problem med funktionaliteten som man inte reflekterat över. Bland annat fanns det ett behov av säkra Internetbetalningar vilket tvingade utvecklarna att inkludera detta. Dessutom behövde man då logga alla aktiviteter som besökarna utförde för att kunna spåra misslyckade eller problematiska betalningar. Detta gjorde att man kallade allt som hände för transaktioner vilka sparades i den helt

ostrukturerade databasen. Vissa delar försvann också helt och hållet då de inte var genomförbara inom utsatt tid. Man avstod till exempel från att utveckla det generella turneringssystemet och sätta upp spelservrar för populära spel. Alla problem bidrog vidare också till att man tvingades ta in ett antal konsulter som fick hjälpa till med utvecklingen.

Trots alla de problem som uppstod i konstruktionsfasen, valde man att inte gå tillbaka och förändra de låsta specifikationer man framställt. *"Det verkade krångligt och dyrt"* (Nils) och *"Jag frågade om de kunde lösa problemen utan att förändra specifikationerna och de svarade ja."* (Christian) Detta ledde enligt Nils till att man ofta löste problem som uppstod med att skriva lite extra programkod som löste problemen. Man försökte desperat trycka in så mycket som möjligt i den ostrukturerade databasen ofta på ställen som var väldigt dåligt anpassade till de aktuella situationerna.

Med tanke på de problem som uppstod så blev projektet kraftigt försenat. Man hade räknat med att vara i drift vid årsskiftet, men var inte klar förrän i slutet av april. Detta ökade kostnaderna så pass mycket att projektet också drog över budget med stora summor.

Då communityn var som mest välanvänd, ungefär två månader efter driftsättningen, fanns ungefär 200 aktiva och betalande medlemmar. Antalet besökare var betydligt fler, men de flesta kom till förstasidan, såg sig om lite och gick direkt igen. Några av de foruminlägg som fanns i det officiella forumet var till exempel:

"I must say that for some of the games I have played with my Trial Premium Membership are pretty pathetic in my opinion and am VERY thankful I won't spend 1 dime on any of them until better games do come out."

Många verkar ändå vara nöjda med funktionaliteten och grundidén med communityn, men menar ändå att det inte är tillräckligt intressant för att locka betalande besökare. Det finns till exempel ett antal funktioner som planerades, men som inte utvecklades vidare. Ett exempel på medlemmars åsikter i denna fråga är:

"Ok, i'm not saying that this is a bad site, more of the opposite. But at least I think that it would be a MUCH better site if it had a chatroom were you don't need to wait hours for answer."

I augusti 2004, när communityn fortfarande bara hade ett hundratal betalande medlemmar, valde ägarföretaget att helt stoppa vidareutvecklingen av systemet. Man såg inte att det fanns någon marknad för denna community och således inte heller någon chans att få rimlig avkastning inom överskådlig tid. Därför valde man att helt och hållet avsluta hela projektet och säga upp de anställda. Kunden var alltså i grund och botten mycket missnöjd med såväl projektet i sig som med projektets resultat, vilket det faktum att de faktiskt stoppade hela projektet visar på.

4.5. Sammanställning

För att man ska kunna få en god överblick över resultatet från de tre fallen bredvid varandra har vi valt att sätta samman en matris med våra undersökningsfrågor på ena kanten och de tre fallen på den andra. Detta ger oss en möjlighet att tydliggöra de data vi fått ur varje fall relaterat till våra undersökningsfrågor.

Tabell 4-1: Sammanställning undersökningsfrågor

Undersökningsfråga	Fall ett	Fall två	Fall tre
<i>Vilken systemutvecklingsmodell hade man som utgångspunkt för utvecklingsarbetet?</i>	En modell som delade upp arbetet i fyra faser (analys, design, konstruktion och test/implementering) låg till grund för arbetet	En sekventiell modell med på varandra följande delar låg som grund för arbetet	En sekventiell modell med analys, design, konstruktion och implementering
<i>Hur påverkade systemutvecklingsmodellen utvecklingsarbetet?</i>	Inledningsvis små resurser ledde till problem i senare skeden som berodde av tidigare resultat. Problem med att det ger en känsla av att man faktiskt kan göra en komplett analys i tidigt skede.	Resursbehovet ökade vid förändring. Svårt att möta förändrade förutsättningar. Kunden involverades inte i tillräckligt hög grad. Alltför starkt beroende av ett fullständigt arbete i tidigt skede.	Man låste sig till de specifikationer man skapat tidigare i projektet, vilket gjorde att man hade svårt att göra ett bra system.
<i>Hur väl mötte resultatet kunden förväntningar (Customer Satisfaction)?</i>	Kunden var nöjd med båda användbarhet och funktionalitet.	Kunden var nöjd med användbarhet, och i stort även nöjd med funktionalitet.	Om man ser ägarföretaget som kund var de mycket missnöjda. Slutanvändarna inte heller helt nöjda.
<i>Hur väl höll projektet sig inom givna resursramar (Efficiency)?</i>	Problem med både tid och resurser. Resurser kompenseras för med att leverantören inte fakturerade för all tid.	Problem med både tid och resurser. Resurser kompenseras för med att leverantören inte fakturerade för all tid.	Budgeten drogs över eftersom man fick ta in extra personal. Tidsplanen överskreds med flera månader.
<i>Vilka nackdelar innebar den valda systemutvecklingsmodellen?</i>	<ul style="list-style-type: none"> - Beroende av tidig analys. - Ökat resursbehov - Brist på återkoppling till analys och design - Svårt att möta förändring 	<ul style="list-style-type: none"> - Beroende av tidig analys - Ökat resursbehov - Distansering av kund - Svårt att möta förändring 	<ul style="list-style-type: none"> - Visst arbete med delar som aldrig utvecklades. - Låsning i analys och designarbetet. - För stort tekniskt fokus
<i>Vilka negativa effekter hade nackdelarna på kundens förväntningar?</i>	Inga	Inga	Att man inte utvecklat vissa funktioner gjorde att den största målgruppen (datorspelare) saknade mycket.

<i>Vilka negativa effekter hade nackdelarna på resursbehovet?</i>	- Ökat behov av arbetstimmar - Problem att hålla tidsplanen	- Ökat behov av arbetstimmar - Problem att hålla tidsplanen	Mycket extra personal behövdes för att följa komplexa specifikationer.
---	--	--	--

5. Analys och reflektion

När vi nu presenterat såväl de data som vi samlat in som den teoretiska studie vi gjort kring ämnet är det så dags att försöka redogöra för vår analys och reflektion över våra data. För att underlätta för läsaren så har vi valt att inleda vårt resonemang i samma struktur som avslutar kapitel tre. Där visar vi på fyra grupper av nackdelar som är särskilt betydande enligt andra forskare; Förändringar i systemkraven, Förändringar i förutsättningar, Fokus på dokumentation och Användardistansering. Dessa grupper fördjupar vi oss alltså i nedan med hjälp av både vårt tidigare resonemang från kapitel tre och med data från vår fallstudie.

Efter att vi gått igenom de fyra grupperna av nackdelar och reflekterat över dem ställer vi oss frågan hur betydelsefulla dessa nackdelar är för arbetet. Vi anser att för att man verkligen ska ta hänsyn till nackdelarna och betrakta dem som av avgörande betydelse så behöver man kunna visa på ett konkret negativt inflytande. För att kunna undersöka nackdelarnas direkta påverkan har vi valt att analysera hur dessa har påverkat projektens resultat i de fall vi studerat. Med resultat menar vi då alltså Customer Satisfaction och Efficiency, som vi tidigare nämnt som avgörande delar av resultatet, för att det ska bli överskådligt.

5.1. Förändringar i systemkraven

Som vi nämnde i kapitel tre så talar Kruchten (2002) om att ett av de mest grundläggande problemen med de sekventiella modellerna är att denna gör det felaktiga antagandet att man i början av projektet kan specificera en komplett lista över alla de krav som ställs på systemet. Detta är någonting som vi upplevt att alla våra tre fall har haft stora problem med. Då den definition man satt upp i tidiga faser inte längre gäller på grund av olika förändringar upplever man det som mycket komplicerat och ”jobbigt” att revidera den kravspecifikation eller systemspecifikation som ligger till grund för konstruktionsarbetet. Det faktum att det upplevs som jobbigt beror som vi ser det av två anledningar. Dels eftersom, som David menar, man upplever det som ett riskmoment. Om man måste gå tillbaka och göra om specifikationen riskerar man ju att upptäcka andra saker som också måste åtgärdas vilket innebär merarbete. Överlag verkar det som att förändring av grundläggande förutsättningar för arbetet känns osäkert och problematiskt. Det verkar vara mentalt jobbigt att gå ”tillbaka” och ändra en färdig specifikation eftersom det insinuerar att man gjort ett dåligt jobb eller att något är fel. Vidare så ser vi ytterligare en anledning i det att det är resurskrävande att lägga tid som skulle ha lagts på konstruktion av funktionalitet till att arbeta om någonting som redan är klart och som högst troligen redan har utnyttjat sin planerade kvot av resurser.

Här tycker vi oss se att det blir ännu mer problematiskt när det inledande arbetet inte utförs med tillräckligt hög kvalitet av olika anledningar eller om man missar att involvera kunden i tillräckligt hög grad. Kunden har ju också åsikter och förhoppningar kring det framtida systemet som om de inte inkluderas i de tidiga fasernas arbete kommer att ge negativa konsekvenser i slutfaserna när produk-

ten testas. Man blir alltså väldigt beroende av arbetet i varje fas eftersom, liksom en dominoeffekt, negativa effekter fortplantas genom påföljande faser. Detta blir speciellt relevant när man studerar forskare som till exempel Kruchten (2002) som påstår att denna utförliga specifikation av grundförutsättningar i tidiga skeden är dömt att innehålla brister eftersom man inte kan uppskatta alla behov eller krav förrän man inlett arbetet med att skapa det faktiska systemet, just eftersom kraven alltid har en tendens att förändras.

Sätter man detta behov av förändring av kraven i relation till den graf över kostnader för förändring relaterat till när i projektet man utför dem som vi visar i del 3.3.1 ovan ser man att det kräver mycket resurser. Detta bekräftar också i vår fallstudie där framför allt fall ett visar på ett stort problem att hålla sig inom givna resursramar när förändringar i kraven på systemet uppenbarade sig. När kundföretagets VD insåg att man byggt upp fakturorna på en felaktig grund, och förändrade denna, så innebär det att hela den struktur och systemspecifikation som man tagit fram och fastslagit inte längre fungerade. Man fick då lägga mycket resurser på att gå tillbaka och göra om strukturen och en hel del av konstruktionsarbetet. Hade man istället arbetat i mindre iterationer utan att fastslå en färdig systemspecifikation så hade det varit betydligt lättare att gå in och modifiera densamma.

Eftersom utvecklingsprojekt ofta har fasta resursramar så innebär detta att projekten drar över dessa med potentiellt mycket negativa effekter som följd.

Vi ser alltså att användarnas behov kan komma att förändras under tiden som utvecklingsförloppet tar. Denna risk ökar såklart med projektets tidsomfattning. Spänner projektet över flera år så innebär det att användarna kommer att utvecklas samtidigt som systemet gör det. Om man då löpande arbetar med en evolutionär process menar vi att man lättare kan utveckla systemet i relation till användarnas utveckling. Detta blir ännu tydligare när man ser till problemet med skillnaden mellan vad användarna tror att de vill ha och vad de vill ha när de ser olika lösningar på sina problem. Kundens VD i fall ett insåg inte vad det var man behövde förrän han passerat över en tröskel och fått se konsekvenserna av den första specifikationen. Det är sällan populärt att då som leverantör komma med någon form av WYSIWYG (what you see is what you get) resonemang. Kunden vill ha en bra lösning på sina problem. Det är som Gulliksen & Göransson (2002) uttrycker det så att man ofta inte är tillräckligt insatt i problemområdet förrän man kommit längre fram i utvecklingsprocessen.

5.2. Förändringar i förutsättningarna

Som vi nämner i vår studie av tidigare forskning så är det inte bara förändringar i krav som påverkar projektets resultat. Många andra faktorer, som man kanske alltid inte reflekterar över, finns och påverkar också projektet.

Ett exempel som projektgruppen i fall ett fick erfara var att den tekniska plattform som man hade valt inte kunde användas till det som man trodde. Brister i Internet Explorer gjorde att man var tvungen att rendera fakturor i Acrobat Reader istället. Detta innebär i sin tur att man var tvungen att omarbete den systemspecifikation som man hade arbetat efter, trots att den var färdigställd och lagd till handlingarna. Ännu värre är det förstås för projekt som löper över flera år när den plattform man valt och bundit upp sig mot kanske blivit undermålig med tiden. Den standard man anammat kanske inte är en standard längre när man är klar.

En annan förutsättning som kan förändras är tillgången på liknande produkter. Om projektet spänner över tiden så ökar ju risken att någon annan utvecklar ett liknande system som löser samma problem, som kunden hellre köper än fortsätter med projektet. Detta är dock ingenting som man konfronterades med i våra fall och det är heller ingenting som egentligen bara är ett problem med de sekventiella modellerna. Det vi vill säga är dock att man genom en evolutionär modell bättre står rustad för att snabbt börja lösa kundens problem med prototyper av det framtida systemet och därför bättre kan konkurrera med andra system på marknaden.

Ett projekt, och då naturligtvis även ett systemutvecklingsprojekt, är som vi nämnt ovan ett fenomen begränsat i tiden och med en begränsad mängd resurser till sitt förfogande. Vi vill mena att projektet så gott som alltid har någon form av tidsplan och budget som specificerar när man bör vara klar och med vilka resurser man förväntas lösa uppgiften. Dessa ramar kan antingen vara hårda eller mer flexibla men faktum kvarstår: Tanken är att man ska lösa problemet med tillgängliga resurser. Det kan andra delar hos kundens verksamhet vara beroende av. Om man till exempel drar över mycket i kostnad kan ju kunden få problem med likviditeten på kort sikt och kanske med lönsamheten på lång sikt.

Om dessa resurser förändras under projektets gång, kanske på grund av att kunden inser att man har ett kritiskt behov av ett fungerande system tidigare än beräknat eller om kunden får problem med resurser antingen i form av pengar eller av annat måste utvecklingsprojektet förändras därefter. Har man en evolutionär modell som grund för arbetet får man här mindre problem eftersom det är lättare att förändra projektets mål då man hela tiden arbetar parallellt med analys, design och konstruktion. Värre är det då om man har låst fast sig i den sekventiella modellen och måste förändra färdiga specifikationer och därmed riskerar att hamna i en situation där man har en förändrad specifikation som inte längre överensstämmer med det man hittills har producerat i senare faser av projektet. Detta blir värre ju längre man kommer från den fas man behöver ändra i, i enlighet med den kurva över resurskostnad för förändring under ett sekventiellt projekt som vi visade i kapitel tre.

Detta drabbade dock inget av våra fall så vi kan inte underbygga resonemanget med resultat ur fallstudien. Vi känner dock ändå att vi vågar, med andra forskares stöd i ryggen, peka på detta som en tämligen stor nackdel.

Slutligen vill vi nämna en kanske lite mer abstrakt, men nog så avgörande, resurs: tillgången på ledningen stöd för projektet. Som Andersen et al (1994) pekar på så kan det vara avgörande för utfallet i projektet. Detta är också föremål för förändring som vi nämner i vår studie av andra forskares arbete i kapitel tre, då till exempel nyckelpersoner i en ledningsgrupp kan ersättas av andra, få minskad makt el dyl. Här står man med en sekventiell modell sämre rustad eftersom man då har en mindre möjlighet till smärtfri förändring i enlighet med de nya direktiv och önskemål som detta kan få som följd. Det blir helt enkelt kostsammare att förändra förutsättningarna för projektet, vilket vi visat på tidigare. Det kan ju i sin tur i värsta fall leda till att de nya personer som fått inflytande inser att detta ju knappast kan vara lönsamt att arbeta med, med följden att projektet läggs ned.

I fall tre ser vi hur ledningens stöd förändrades då man inte tyckte att man fick tillräckligt med resultat, någonting som kanske skulle kunna ha undvikits om man arbetat efter en evolutionär modell med prototyper istället. Då skulle man ha kunnat nå konkreta resultat snabbare vilket kanske skulle ha kunnat lugna ledningen så att de inte hade behövt lägga ned hela projektet som ju blev följden nu.

5.3. Fokus på dokumentation

Att man i ett sekventiellt uppbyggt systemutvecklingsprojekt skulle fokusera mycket på att framställa dokumentation kan vid första anblicken verka som ett rimligt antagande. Sommerville (2000) kallar till och med de modeller, som vi grupperar under samlingsnamnet sekventiella, för specifikationsbaserade. Detta för att varje fas bygger på en specifikation som är resultatet av den föregående. Vi har dock till vår egen förvåning upptäckt att detta inte gäller för de projekt som vi undersökt i vår fallstudie. I dessa gjorde man ganska övergripande och löst hållna kravspecifikationer som togs vidare till ganska enkla systemspecifikationer, om man ens gjorde någon överhuvudtaget. Det enda projektet där man genomförde en ordentlig systemspecifikation (men fortfarande en mycket vag kravspecifikation) var fall tre där man byggde upp ett mycket tekniskt komplext UML-diagram. Man måste dock inse att bara ett tekniskt komplext UML-diagram enligt oss inte räcker för att klassas som dokumentationsintensivt.

Det som dock slår oss när vi tittar på detta är att man inte bara producerade lite dokumentation utan också hade en motvilja mot att gå tillbaka och ändra den. De mycket löst hållna kravspecifikationerna hölls alltså hårt på, precis som de nästan obefintliga systemspecifikationerna. I fall tre ledde den onödigt komplexa, men ändå mycket otillräckliga, systemspecifikationen till att man fick en systemarkitektur som tyvärr var helt ohållbar att arbeta i. Detta var något man insåg här men ändå var man oviljig att gå tillbaka och göra förändringar i specifikationen. Som Nils säger: *"Vi skapade en alldeles för teknisk komplex design som inte alls överensstämde med det ändå ganska enkla communityt. Till slut blev det helt ohållbart att göra förändringar i systemet."* Denna kombination vill vi mena är flera gånger värre än att göra utförlig och framför allt genomarbetad dokumentation som man fryser. Hade dokumentationen bara varit genomarbetad, så hade man antagligen haft många problem, osäkerheter och felaktigheter i specifikationerna men hade antagligen klarat sig betydligt bättre än man nu gjorde, i princip helt utan dokumentation.

Vari ligger då denna motvilja att gå tillbaka och förändra tidigare låsta specifikationer. Grundtanken i vattenfallsmodellen är att man maximalt ska gå tillbaka ett steg, då det annars skulle bli alldeles för dyrt. Detta är också något som kommer igen i alla de fall vi studerat. I fall ett ser vi till exempel att man tyckte att det var komplicerat och jobbigt medan Nils menar att *"Det verkade krångligt och dyrt"*. Detta är intressant, särskilt i fall tre där det antagligen hade blivit en mindre kostnad att gå tillbaka och göra om systemspecifikationen från grunden när man upptäckte hur felaktig den var än vad det faktiskt blev när man inte gjorde det. Det vi anser oss ha märkt här, är att det verkar som om att, även om man har möjlighet att gå tillbaka till tidigare faser, så ger detta sätt en inneboende motvilja mot att återkoppla, vilket också Gulliksen & Göransson (2002) håller med om. Egentligen är detta inte konstigare än att de flesta har en motvilja mot att göra fel. Att återkoppla till en tidigare fas skulle, som vi nämner ovan, kunna vara att erkänna att man gjort fel vilket man kanske inte alltid är så bra på att göra.

5.4. Tekniskt fokus och användardistansering

Vissa forskare, som Gulliksen & Göransson (2002), menar som vi nämnt tidigare att dokumentationen ofta är tekniskt komplex, med påföljden att man tenderar att förlora kontakten med slutanvändarna. Detta i sin tur leder till en risk att det slutgiltiga systemet blir mindre användarvänligt och att man som kund därmed får ut mindre ur sitt utvecklingsprojekt än om man hade haft en tydligare medverkan från sina användare i utvecklingsprocessen.

Detta hänger ihop med att man, enligt Gulliksen & Göransson (2002) även riskerar att i ett systemutvecklingsprojekt med en sekventiell utvecklingsmodell som grund få en alltför teknisk inriktning på arbetet. Detta leder också till att man ytterligare riskerar att distansera sig från användarna eftersom dessa inte förstår den dokumentation som produceras och inte heller vad det pratas om på möten osv. Den stackars användaren, som inte riktigt kan formulera sina behov i enlighet med det språk som de andra i utvecklingsgruppen brukar, ses då som ett riskmoment. Han eller hon kan när som helst inse att det som alla kommit fram till inte betydde vad användaren trodde och därmed ändra sig, med mer arbete för utvecklarna som följd. Uttrycket att ”Användare vet aldrig vad de vill” behöver alltså inte betyda att användare överlag skulle vara lite bakom.

I vårt tredje fall ser vi tydligt fokus på att utveckla en tekniskt väl fungerande produkt av hög kvalitet. Detta innebär dock att man tyvärr distanserar sig från användarna i det att deras behov, alltså bland annat systemets användbarhet, blir mindre viktiga. Ett bra exempel är gränssnitten som man beställde från ett annat företag och implementerade mer på premisserna att det såg bra ut och att det passade väl in i den tekniska specifikationen än på att det skulle vara testat på användarna och verifierat användarvänligt.

Även i fall två ser vi hur användarna av olika skäl distanseras från utvecklingsarbetet med påföljden att de inte är medvetna om hur arbetet fortlöper. De kastar istället glatt in nya direktiv i ett projekt som allt sämre klarar av att leverera resultat på ett fullgott sätt.

Vi skulle här vilja kunna säga att detta är sanningen; systemet kan eventuellt bli jobbigare att ta fram med hög användarmedverkan men blir då istället ett system som möter rätt problem på ett korrekt och tillfredsställande sätt. Vi kan dock inte se några avgörande tecken på detta i vår fallstudie. Enligt vad vi kan se så är kunderna i fall ett och två, trots att det förekommit brister med användardistansering i stort nöjda med projektens resultat. De upplever att de har fått ett användbart system som löser de behov som de hade. I fall tre ser vi dock att kunden varit betydligt mindre nöjd med resultatet, till och med i den höga grad att man har lagt ner hela projektet.

5.5. Nackdelarnas påverkan på resultatet

För att försöka redogöra för om huruvida dessa nackdelar som vi lagt märke till faktiskt utövade någon signifikant negativ påverkan på resultatet i de fall vi studerat presenterar vi nedan en analys av detta.

Med resultatet menar vi alltså de två specifika delar av vad man kan kalla ett systemutvecklingsprojekts resultat som vi redogjort för ovan och med nackdelar menar vi de fyra olika grupperna som vi presenterat.

Inledningsvis vill vi visa de direkta samband som vi hittat i våra fall med hjälp av ett påverkansschema som man kan se som en mycket förenklad variant av Miles & Hubermans (1994) kausala nätverk. Som entiteter har vi våra fyra kategorier av nackdelar som de två valda delarna av resultatet. Med hjälp av pilar visar vi, utan att värdera eller gradera, påverkan.

Vad gäller den påverkan som "Förändringar i systemkrav" utövade så såg vi i fall två en tydlig problematik med att kunden förändrade kraven på systemet löpande. Detta upplevde de som arbetade i projektet som mycket problematiskt och jobbigt eftersom man hela tiden var tvungen att antingen gå tillbaka och revidera all inledande planering eller bara helt enkelt "koda och fixa" så gott det gick. Nu valde man dessutom alternativ två vilket innebar att resultatet inte blev så gott som man hade hoppats på. Även i fall ett så såg vi denna typ av negativa påverkan. Vad innebar då detta för resultatet? Vad gäller Customer Satisfaction i fall två så ser vi att trots att användbarheten är god så saknas vissa delar av den önskade funktionaliteten fortfarande. Detta på grund av att man helt enkelt inte lyckade inkludera all den nya funktionaliteten på ett bra sätt utan den lades till ryckigt och sällan, enligt respondenterna, väl genomtänkt. Även i fall ett ser vi att kunden inte är helt nöjd med funktionaliteten även om användbarheten återigen tycks vara god. Problemet är dock inte lika omfattande som i fall två.

Påverkan på Efficiency är tydligare. Här ser vi både i fall ett och i fall två att denna grupp av nackdelar haft stor påverkan. Tidsplaner har inte kunnat hållas och resursramarna har överskridits. Att man har kompenserat för detta genom att inte fakturera för all upparbetad tid förändrar inte det faktum att man har stora problem. Vi ser alltså sammantaget att "Förändringar i systemkrav" påverkar både Customer Satisfaction och Efficiency emedan påverkan på Efficiency är tydligare och kraftigare.

Gruppen av nackdelar som vi valt att kalla "Förändringar i förutsättningar" ser vi också tecken på i våra två första fall, främst då i fall ett där bland annat den tekniska plattform som man valt inte höll måttet när det visade sig att man inte kunde få fakturor på flera sidor än en. Detta ledde till problem i och med att man nu fick förändra mycket i arbetet och välja en ny plattform. Dessa nackdelars konsekvenser är dock inte lika tydliga när det gäller att relatera dem till våra två delar av resultatet. Vad gäller Customer Satisfaction så ser vi inte att det skulle ha någon som helst påverkan, kunden har inte varit intresserad av processen utan endast av slutprodukten. När vi sedan flyttar blicken till Efficiency ser vi dock en negativ påverkan främst relaterat till den stora resurskostnad som uppkommer när man utför en förändring i ett sekventiellt projekt som hunnit arbeta sig in i sina senare faser (se 3.3.1 ovan). Detta blev främst ett problem i fall ett där man som leverantör fick investera stora resurser i att byta teknisk plattform mitt i projektet, men även i fall två ser vi denna negativa påverkan.

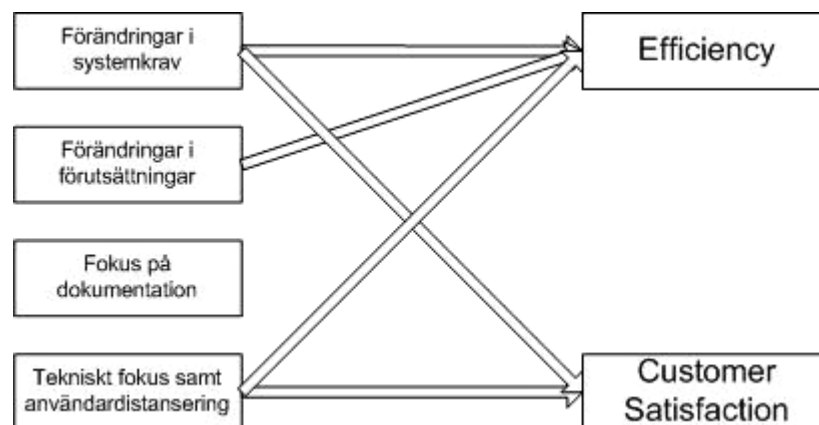
Den tredje gruppen av nackdelar, "Fokus på dokumentation", ser vi som vi nämner ovan inte mycket av i våra fall. Vi ser alltså ingen påverkan i våra fall vare sig på Customer Satisfaction eller på Efficiency. Huruvida detta kan tänkas gälla för alla systemutvecklingsprojekt som baseras på en sekventiell modell kan vi dock inte svara på, då graden av dokumentation också kan påverkas av andra faktorer.

När vi kommer till den fjärde gruppen nackdelar ser vi dock en negativ påverkan. "Tekniskt fokus samt användardistansering" skapar problem både i fall två och i fall tre. I fall två ser vi främst ett

problem relaterat till Efficiency eftersom man, då man distanserat sig från slutanvändarna och kunden, helt verkade sakna en förståelse från kunden för problem och komplikationer i arbetet. Kunden bara fortsatte att ålägga projektet med nya önskemål utan en tanke på den negativa inverkan detta skulle få på resultatet. Vi ser dock även hur detta blir relaterat till Customer Satisfaction då slutanvändarna inte är helt nöjda med funktionaliteten. Eftersom de hela tiden tilläts fortsätta komma med nya önskemål utan att förstå hur detta inkluderades i systemet förväntade de sig någonting som inte överrensstämde med verkligheten. Detta skapade en obehaglig situation när systemet skulle sluttestas inför leverans.

I fall tre ser vi framförallt hur man genom att fokusera på den tekniska processen distanserat sig från användarna med påföljden att dessa inte kände att de kunde påverka vad som upplevdes vara en undermålig produkt. Det ledde i sin tur till att användare av systemet inte strömmade till i den grad som kunden, alltså moderbolaget, hade förväntat sig. Detta tror vi mycket väl kan vara en av anledningarna till att man lade ned verksamheten.

För att tydligare åskådliggöra vad vi nu resonerat oss fram till visar vi nedan en summering av de påverkansförhållanden som vi sett. En pil betyder att vi här har sett någon form av negativ påverkan i vår fallstudie. Summerat ser det alltså ut som i Figur 5-1 nedan:



Figur 5-1 Översikt över nackdelarnas påverkan på resultatet

Vi vill dock inte nöja oss med detta. För att slutligen ge läsaren möjlighet att kunna bilda sig en uppfattning om nackdelarnas påverkan på resultatet vill vi även ge en översiktlig bild av graden av negativ påverkan. Detta illustrerar vi genom Tabell 5-1. Graden av påverkan baseras på den diskussion vi just förde och skalan är uppdelad enligt följande:

Skala:

Ingen påverkan	
Låg påverkan	*
Medelhög påverkan	**
Hög påverkan	***

Ingen påverkan talar för sig själv. Med låg påverkan avser vi fall där vi sett tecken på negativ influens men inte kan se att det skulle ha påverkat i nämnvärt hög utsträckning. Med medelhög påverkan menar vi de fall där den negativa influensen är mer tydlig och där man även kan se att det spelat viss roll för resultatet. Slutligen så menar vi med hög påverkan de fall där vi sett hur nackdelarna och den negativa påverkan de fört med sig kraftigt har påverkat resultatet i utvecklingsprojektet.

Nackdelar relaterade till:	Påverkan på Customer Satisfaction			Påverkan på Efficiency		
	Fall ett	Fall två	Fall tre	Fall ett	Fall två	Fall tre
Förändringar i systemkrav	*	**		***	***	
Förändringar i förutsättn.				**	*	
Fokus på dokumentation						
Användardistansering		*	***		**	

Tabell 5-1: Grad av nackdelarnas påverkan på resultatet

Till detta vill vi slutligen lägga det faktum att även om nackdelarnas påverkan på Efficiency inte skulle väga lika mycket som påverkan på Customer Satisfaction (som ju var den mest betydande delen av resultatet) så är de i och med Efficiencys påverkan på Customer Satisfaction klart värda att ta med i eventuella riskkalkyleringar.

5.6. Finns det fler nackdelar?

Vi vill även passa på att resonera kring andra eventuella nackdelar än de som tas upp i vår teoretiska studie i kapitel tre. Vi ser framförallt två andra typer av nackdelar som vi inte kan finna i litteraturen men som vi tycker är tydliga i vår fallstudie. Dessa två typer är problemet med bristande kompetens hos projektdeltagarna och den låga insyn som personer externt från projektgruppen har. Vi vill mena att dessa problem förvärras i och med det sekventiella flödet av aktiviteter i utvecklingsarbetet.

Brister i kompetens hos deltagarna i projektgruppen, och då menar vi allmänna brister relaterat till kunskap och erfarenhet av generell systemutveckling, riskerar att slå särskilt hårt mot sekventiella modeller då man inte stämmer av särskilt ofta mot kund. Man märker inte av att en fas har problem förrän man avslutat den och gått vidare till nästa fas, som ju ofta bygger på den förra fasens resultat. Detta förvärrande av bristen på generell kompetens ser vi kanske framför allt i våra två första fall och vi tycker det bitvis är väldigt tydligt hur denna brist förvärras av upplägget av arbetet. Den brist på kompetens vi talar om i detta avsnitt skiljer sig från den vi berör nedan relaterat till problemet med de evolutionära utvecklingsprocesserna (avsnitt 6.1.1) i det att det inte är en brist på kompetens relaterat till en specifik utvecklingsprocess eller modell utan en mer allmän sådan som kan bero på till exempel bristande erfarenhet av systemutvecklingsarbete eller brist på adekvat utbildning.

Detta faktum att man inte stämmer av ofta mot kund är även en del i det andra problemet, det med låg insyn, som gör det svårt för ägare, kunder eller andra externa personer att se vart projektet är på väg. Man levererar ju väldigt lite resultat när arbetet befinner sig i en fas, det är snarare i avslutandet av faser som dokumentation sammanställs och eventuellt levereras ut till externa personer. Enligt vår erfarenhet levereras väldigt lite information till kund förrän sent i projektet. Effekten av detta ser vi framför allt i vårt fall två ovan där kunden trott att projektet skulle resultera i någonting annat än vad det gjorde, med stora förändringar och problem som följde.

Vi vill mena att detta skiljer sig från problemet med användardistansering i och med att det inte bara är från användarna som det är problematiskt att distansera sig. Resterande delar av kundorganisationen såväl som delar i den egna organisationen som påverkas av projektet är såklart intresserade av ett visst mått av insyn och kontroll. Risken är att de synpunkter och önskemål om förändringar som viktiga externa grupper har på utvecklingsprojektet kommer in för sent eftersom de har svårt att se var man ligger i utvecklingsarbetet. Relaterat till problemen med förändring vi talat om tidigare vill vi mena att detta kan bli riktigt problematiskt. Detta skulle enligt vår mening skilja sig från de evolutionära utvecklingsmodellerna i vilka man arbetar i korta iterationer och därför blir det också naturligt med många mindre avstämningar efter varje iteration. Många evolutionära utvecklingsprocesser, som till exempel XP (Beck, 2001), betonar just viken av regelbundna avstämningar och ett nära samarbete med kundorganisationen.

6. Avslutande diskussion och presentation av resultat

Nedan ämnar vi försöka knyta ihop säcken som är vår analys och diskussion samt ur detta dra fram och presentera våra resultat. Vi gör även en kort diskussion där vi försöker att kritiskt granska vårt eget arbete. Slutligen visar vi på möjligheten till vidare forskning inom samma ämne.

6.1. Att knyta ihop säcken

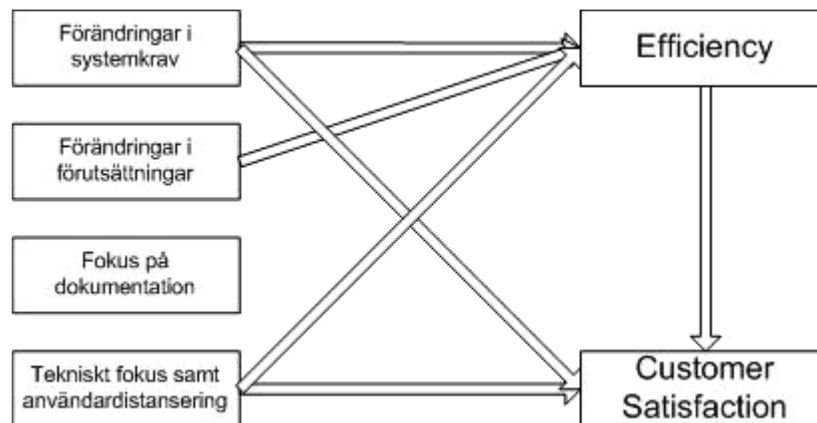
Vi ser som vi visar i tabell 5-1 en tydlig negativ påverkan vad gäller framför allt förändringar i systemkrav och användardistansering. Viss negativ påverkan återfinns vid förändringar i förutsättningar emedan fokus på dokumentation inte alls verkar påverka relevanta delar av resultatet negativt.

6.1.1. Grad av påverkan på resultatet

Man kan klart tycka att den negativa påverkan på Customer Satisfaction inte är särskilt hög och att detta då skulle vara ett argument för att det inte skulle vara ett större problem för projektets resultat. Vi vill här påminna om det resonemang vi fört om resultatet ovan där vi refererar till forskare som menar att Customer Satisfaction vida överskuggar de andra delarna av resultatet.

Dessutom ställer vi oss frågan om man egentligen kommit tillräkta med användardistanseringen om man använt någonting annat än en sekventiell modell som grund för sitt arbete. Även om författare som Beck (2001) menar att man utvecklar i nära samverkan med de framtida användarna så menar vi att man när man påstår detta förutsätter att utvecklarna är experter på den valda utvecklingsmodellen och följer denna helt enligt boken. Detta är enligt vår egen erfarenhet (Bendelius, Jönsson, Olsson, 2003) inte alltid gällande. Författarna såg i sitt arbete att det var lätt att man som utvecklare tolkade komplicerade avsnitt i utvecklingsprocessen så det passade det egna arbetssättet och framförallt förenklade budskapet med en mer rätlinjig process som följt. Detta ledde enligt författarna även till ett sämre resultat.

Med detta kan man ju tycka att påverkan på den mest betydelsefulla delen av resultatet då bleknar när den enda kritiska faktor som vi hittat försvagas. Värt att nämna här är dock en annan typ av påverkan som vi har hittat i vår fallstudie, nämligen den påverkan som den ena delen av resultatet, Efficiency, utövar på den andra, Customer Satisfaction. Hur nöjd kunden är med projektets utfall är enligt våra fall klart relaterat till hur väl projektet i övrigt håller sina mål såsom tidsplaner och resursramar. Vi tycker att det är naturligt att ett projekt som drar över tiden eller som spräcker sin budget torde ha svårare att få lika nöjda kunder som projekt som håller sig inom givna ramar och förutsättningar. Den nya bild som detta resonemang genererar visar vi i figur 6-1 nedan:



Figur 6-1 Slutlig påverkansmodell

I och med detta så menar vi att den negativa påverkan på resultatet i och med påverkan på Customer Satisfaction är större än vad man kan tro om man bara ser till den tidigare modellen vi presenterade. Därav är alltså de nackdelar som vi tar upp med de sekventiella utvecklingsmodellerna i förlängningen en källa till ett sämre resultat. Med detta underbygger vi det resonemang vi fört kring att de nackdelar som vi tar upp med de sekventiella utvecklingsmodellerna i förlängningen är en källa till ett sämre resultat.

6.1.2. *Varför väljer man då en sekventiell modell?*

Väljer man överhuvudtaget en modell? Vi vill mena att man i våra fall varken aktivt valt en modell eller en process. Vi har istället fått utgå från det arbetssätt som man valt och hur man i detta lagt upp de olika delarna i arbetet för att försöka skapa oss en bild av den omedvetna modell som legat till grund för arbetet.

Detta att man inte medvetet arbetar efter en specifik utvecklingsprocess är inte någonting som är unikt för våra fall. Det finns tidigare undersökningar (Russo & Stolterman, 2000) som visar att många projekt inte arbetar efter en process, eller i alla fall inte efter någon etablerad sådan. Istället grundar man sitt arbete på tidigare erfarenheter och på ett inom gruppen etablerat arbetssätt. (Russo & Stolterman, 2000)

Frågan blir då snarare varför man omedvetet bygger sitt utvecklingsarbete på ett sekventiellt flöde av aktiviteter? En anledning kan kanske vara att ett sekventiellt arbetssätt upplevs som enkelt och okomplicerat samt att det liknar många andra klassiska aktivitetsflöden som till exempel hur man går till väga när man bygger hus. Här bestämmer man sig först för vad man ska ha, sedan ritas en arkitekt upp ritningarna och slutligen så bygger man huset. Det upplevs enligt vår erfarenhet och enligt de svar vi fått av våra respondenter som ett naturligt sätt att arbeta.

Vidare kan man spekulera i utvecklarnas utbildnings påverkan på arbetssättet. Enligt vår erfarenhet av akademisk grund för systemutveckling vid Institutionen för Informatik (Lunds Universitet, åren 2000-2005) grundar sig denna ofta på sekventiella faser som grund för utvecklingsarbetet. Vi ser även en möjlighet i att man som ung nyutbildad systemutvecklare kanske påverkas av andra erfarna kol-

legor som eventuellt har sin akademiska och yrkesmässiga grund i en tid innan man inom forskningsvärlden började ifrågasätta de sekventiella modellerna.

Vi ser alltså att man i våra fall inte arbetat enligt en systemutvecklingsprocess och man har inte heller aktivt valt en sekventiell modell som grund för arbetet. Detta gör problemet svårare att komma åt. Man följer inga formella riktlinjer eller regler som man kan ändra på. Det är betydligt svårare att påverka människors attityder och åsikter om vad som är ett naturligt och bra sätt att arbeta än att förändra en formell process.

6.2. Slutsatser

Vårt arbete bygger som bekant på den grundläggande frågeställningen ”Vilka nackdelar finns, relaterat till resultatet, med att grunda arbetet i ett systemutvecklingsprojekt på en sekventiell systemutvecklingsmodell?” och vi kommer här att sammanställa våra slutsatser.

I vår teoretiska studie ser vi nackdelar relaterat till att möta förändring. Detta gäller både förändring i krav på och förändring i förutsättningar för projektet. Detta ser vi tydliga spår av i vår fallstudie som alltså underbygger de resonemang vi fört i kapitel tre. Ett systemutvecklingsprojekt baserat på en sekventiell utvecklingsmodell riskerar att få problem om krav eller förutsättningar förändras i och med den motvilja att gå tillbaka och förändra tidigare arbete som vi ser.

Vidare ser vi också ett problem med att man i och med ett sekventiellt arbetssätt distanserar sig från användarna. Detta leder till att man riskerar att användarna inte får grepp över det framtida systemet förrän i ett sent skede av projektet vilket då kan leda till snabba förändringar vilket man då enligt vårt resonemang ovan är dåligt rustad att möta.

Vi ser däremot inte spår av nackdelar relaterat till fokus på dokumentation vilket gör att vi inte kan underbygga det resonemanget med vår fallstudie. Dock ser vi istället två andra typer av nackdelar som vi inte kan finna i teorin. Det första vi ser är att ett arbete baserat på en sekventiell utvecklingsmodell riskerar att förvärpa effekterna av eventuella brister i den generella kompetensen hos projekt-deltagarna. Den andra och sista typen av nackdel vi ser är den bristande insyn som ett sekventiellt arbete riskerar att ha. Få avstämningar gör att externa personer, och då menar vi primärt inte de framtida användarna, får svårt att greppa projektets status.

Avslutningsvis har vi kompletterat detta med ett resonemang kring det ofta omedvetna valet av en sekventiell modell som grund för arbetet vilket riskerar att minska möjligheten att påverka detta val. Det finns ofta inget specifikt val av process som man kan förändra utan det rör sig snarare om en syn på sekventiella modeller som ett naturligt sätt att lägga upp arbetet.

Vårt resonemang kring att man genom att, ofta omedvetet, välja en sekventiell utvecklingsmodell som grund för sitt arbete drar på projektet ett antal nackdelar som man bör vara medveten om för att säkerställa att deras påverkan på resultatet minimeras är vårt främsta kunskapsbidrag.

6.3. Reflektion över arbetet

Man skulle kunna ifrågasätta vårt val att endast titta på de delar av resultatet som Dvir & Lechler (2004) tar upp. Man skulle kunna tänka sig att andra forskare talar om ett antal andra faktorer som viktiga beståndsdelar av ett projekts resultat, vilket skulle, om vi använt dessa definitioner istället, ge upphov till ett resultat helt skilt från det vi nu har kommit fram till. Vi vill dock mena att Dvir & Lechlers (2004) forskning är väl lämpad att ligga till grund för vår analys, då den ger oss två väl definierade faktorer att relatera modellernas nackdelar till. Dessutom har vi varit tvungna att begränsa oss i vårt undersökningsarbete, vilket lett oss till att endast utgå från de aktuella faktorerna. Man skulle annars kunna tänka sig att man i vart och ett av fallen skulle kunna göra flera analyser utifrån de olika perspektiven på ett systemutvecklingsprojekts resultat. Detta var dock inte möjligt med hänsyn taget till tillgängliga resurser.

Vi kan även se ytterligare begränsningar i vårt arbete med att vi begränsat oss till att i stort sett endast utgå från tidigare forskning när vi letat efter vilka nackdelar som finns i de sekventiella modellerna. Vi har alltså inte lagt mycket kraft på att försöka hitta nya nackdelar med sekventiella utvecklingsmodeller än de som andra forskare har redogjort för, vilket vi skulle ha gjort om vi till exempel istället använt oss av någon form av ”grounded theory” som bas för vår undersökning. Att vi haft vår utgångspunkt i tidigare forskningsresultat ger oss i och för sig en tryggare grund att basera vårt arbete på men vi missar delvis chansen att se andra nackdelar som kan ha påverkat resultatet. Detta hade kunnat vara särskilt intressant i just detta fall då vi inte direkt kan skylla hela de dåliga resultaten i projekten på de nackdelar vi tar upp. Ett mer utförligt arbete med att söka efter andra grupper av nackdelar kunde kanske ha spridit ett skarpare ljus över situationen.

Slutligen ser vi nu i efterhand att vi troligen hade fått ett bättre underbyggt resultat om vi i våra fall även hade intervjuat kunderna. Detta eftersom vi till största delen använder just Customer Satisfaction som grund för att se huruvida nackdelarna påverkat resultatet negativt. Istället har vi valt att intervjua projektdeltagarna om kundens inställning till projektet och dess resultat vilket naturligtvis inte ger samma tyngd som om vi hade intervjuat kunderna själva. Detta är en brist som vi är medvetna om och vi råder eventuell vidare forskning att ta hänsyn till detta.

6.4. Vidare forskning

Den första begränsning vi ser i vår undersökning, det faktum att vi bara tittar på resultatet utifrån ett forskarlags resultat, kan vara en passande ingång för framtida forskare att titta på. Man skulle till exempel kunna utgå från andra forskares teorier om relevanta delar i ett systemutvecklingsprojekts resultat för att på så sätt hitta ytterligare nackdelar med de sekventiella systemutvecklingsmodellerna. Vi visar ju på ett par andra nackdelar som vi ser spår av i vår fallstudie. Dessa kunde med fördel analyseras i framtida uppsatser eller rapporter.

Ett ytterligare förslag på framtida forskning, som dessutom tycks falla sig väldigt naturligt med vår undersökning som grund, är att även titta på vilka nackdelar som eventuellt finns med de evolutionära modellerna. Tveklöst är det så att även de evolutionära modellerna innebär ett antal nackdelar som kan vara intressanta att belysa, då dessa i våra ögon dessutom är än mindre undersökta än de sekventiella modellerna.

7. Bilagor

Fallstudieprotokoll

Datum: _____

1. Introduktion

Vilka nackdelar finns, relaterat till resultatet, med att grunda arbetet i ett systemutvecklingsprojekt på en sekventiell systemutvecklingsmodell?

Syftet med denna undersökning är dels att göra en teoretisk undersökning i vilken vi undersöker hur andra forskare och författare ställer sig till att ha en sekventiell modell som grund för arbetet i ett systemutvecklingsprojekt och dels att underbygga den undersökningen med en **fallstudie** i vilken vi undersöker ett flertal sådana projekt, i vilka man använt sig av ett sekventiellt arbetssätt, för att försöka visa på de eventuella negativa aspekter som detta förde med sig. Detta för att leda oss fram till våra slutsatser som svar på vår frågeställning ovan.

2. StudF – Studenten i fält

Kontaktperson: _____ Telefonnummer: _____

Information hämtas från två olika källor. Dels så skall *intervjuer* genomföras med nyckelpersoner, enligt intervjuprotokoll nedan, dels så skall lämplig *dokumentation* av utvecklingsarbetet studeras. Respondenterna väljs ut med hänsyn till deras involvering i utvecklingsarbetet. Prioritera hög delaktighet.

3. Undersökningsfrågor

- Vilken systemutvecklingsmodell hade man som utgångspunkt för utvecklingsarbetet?
- Hur påverkade systemutvecklingsmodellen utvecklingsarbetet?
- Hur nöjd var kunden med projektets resultat (Customer Satisfaction)?
- Hur väl höll projektet sig inom givna resursramar (Efficiency)?
- Vilka nackdelar innebar den valda systemutvecklingsmodellen?
- Vilka negativa effekter hade nackdelarna på Customer Satisfaction?
- Vilka negativa effekter hade nackdelarna på resursbehovet (Efficiency)?

4. Datahantering

Intervjusvar transkriberas på dator i de fall som anteckningar tagits för hand. De ska dock behålla sin originalstruktur och inte sammanställas i detta skede.

Dokumentation kopieras eller skrivs av så långt som möjligt så att en egen kopia kan erhållas. I de fall så inte är möjligt tas anteckningar kring innehållet.

5. Intervjuguide

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Intervjuanteckningar

Fall ett – ”Kalle”

Projektbeskrivning

Kalle (man, 28 år) arbetar i ett mindre utvecklingsföretag. Under projektet samtidigt deltidsanställd som IT-ansvarig hos kunden. Kund var medelstort konsultföretag som arbetar med konsulter åt läkemedelföretag.

Kunden var inte nöjd med sitt nuvarande system för tidrapportering. Systemet var gammalt, skrivet i Access, och förlegat. Det uppfyllde inte kundens behov. Kunden utvecklades mycket snabbt med ökade behov som följd. Systemet byggde dessutom på en bristande specifikation. Kunden var i behov av ett nytt system.

Utvecklingsarbetet inleddes under 2000. Budget 400 000kr. Projektgruppen bestod av sex personer. Det fanns olika programmerare eftersom det skulle byggas två gränssnitt; webbgränssnittet var primärt ett tidrapport- och faktureringsgränssnitt som användarna skulle jobba i och sedan fanns även ett administratörsgränssnitt (Javaklienten).

En beställning genomfördes i projektets inledning, i huvudsak av Kalle. Dock så stämde alla idéer av med företagsledningen innan projektet inleddes. Planeringen byggde inledningsvis på att all verksamhet byggde på projekt. Fakturering skulle ske på projektbasis.

Ett grundläggande krav på systemet var att det skulle kunna användas oberoende av geografisk plats med tanke på företagets expansionsplaner. Därför utvecklade man en webbaserad applikation.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

Vi arbetade inte direkt efter någon specifik utvecklingsprocess, utan använde en modell som delade upp arbetet i fyra faser (analys, design, konstruktion och test/implementering) som grund för arbetet. Detta kändes som ett naturligt och lätt sätt att lägga upp arbetet. Vi reflekterade nog inte egentligen så mycket över detta utan arbetade efter erfarenhet.

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

Arbetet som helhet påverkades en del av upplägget i faser. Det avsattes små resurser inledningsvis till analys och design. Dessutom användes resurserna på ett felaktigt sätt. Till exempel förväntades systemet vara modellerat på en vecka, vilket skapade stor tidspress. Detta berodde på att systemets storlek underskattades och för att man trodde att det var enbart databasen som egentligen var i behov av design. Detta ledde bland annat till att funktionaliteten i gränssnitten inte designades med helt

oläsbar kod som följd. Detta insåg inte Kalle på grund av att han inte deltog i själva kodarbetet. Det var först när man försökte göra anpassningar som Kalle insåg att koden var omöjlig att underhålla och omöjlig att utveckla vidare. Detta ledde senare till att kodbasen var tvungen att skrivas om i nästa projekt.

Uppgiften påverkade behovet av en återkoppling till tidigare delar, i och med att systemet var tämligen omfattande. Speciellt med tanke på projektgruppens storlek och budget. Man hade nog tjänat på ett mindre sekventiellt upplägg, i det att man då slapp stänga tidigare faser.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Upplägget av aktiviteterna skapade till viss del en motvilja att återkoppla till tidigare arbete. Man ville inte behöva se att man inte skulle ha gjort rätt, att man inte skulle ha konstruerat systemet på ett fullgott sätt. Återkoppling skulle ju innebära att man måste ta konsekvenserna av vad man skulle kunna upptäcka. Det kunde även ses som beroende på ren lathet. Man orkade inte åtgärda fel som kunden inte hade märkt. ”*Om de inte märkt att det är finns fel så finns det inga fel*” var en stark uppfattning hos vissa av deltagarna.

Det kan dock även vara så att det är en kombination av brist på resurser som tid, pengar och motivation. Det var lättare att anamma en sekventiell modell eftersom det verkade lättare att få en översikt och dessutom verkade ständiga iterationer vara kostsamma. Det innebar att användarna involverades främst i början och i slutet av projektet. Det ledde dock enligt Kalle inte till att distansera utvecklingsarbetet från systemets framtida användare. Detta eftersom löpande genomgångar med användarna genomfördes ända fram slutttestningen av systemet vilket ledde till att användarna ändå fick en känsla av delaktighet.

Under arbetet var Kalle marginellt medveten om att återkopplingen brast, det är en lärdom som han känner att man kan dra ur processen samt ur resultatet. Om man hade haft bättre koll så hade man sett till att avhjälpa situationen. De andra deltagarna hade ännu mindre koll på vikten av återkoppling. Detta tror Kalle främst berodde på att ingen i projektet hade någon kunskap kring detta med sig i bagaget.

Dokumentationen påverkades inte av aktiviteternas upplägg.

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Systemutvecklingsmodellen var inte alls av intresse för kunden. De tyckte att de hade ett tydligt behov som man utan problem kunde dokumentera i de inledande faserna av projektarbetet. Man såg

inte att det skulle kunna uppkomma några större framtida behov och ville inte lägga resurser på att till exempel återkoppla till analys och design. Detta till trots att Kalle även var beställare och hade inflytande på detta. Detta berodde dels på en oförmåga hos Kalle att förmedla vikten av en förmåga att hantera förändring i krav och förutsättningar, dels på beställarens dubbla roll (Kalle tappade i trovärdighet när han både var anställd samt arbetade som extern konsult) samt även av okunskap hos kunden. Kunden hade bristande insikt i systemets omfattning och i betydelsen av systemet för sin verksamhet. Man påverkades även av ekonomiska faktorer.

Kalle anser att kunden på det hela taget var nöjd med funktioner och användbarhet. Dock så var de troligen inte riktigt lika nöjda med att projektet kraftigt hade överskridit sin tidsplan. Det brukar ju inte direkt höra till de saker som man som kund är överförtjust i, och detta var nog inget undantag. Kunden gjorde dock inget större väsen av detta utan var ändå nöjda med att de till slut fick sitt system.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Projektet lyckades egentligen varken hålla sin tidsplan eller sina kostnadsramar. Detta eftersom det var problem under arbetet.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Det visade sig efter ett par månaders arbete att beställningen och planeringen var orealistiska på grund av att företagsledningen inte hade koll på innebörden och konsekvenserna av att ha projektredovisning som bas för systemet. Man hade tydligt angivit att all verksamhet byggde på projekt. Fakturering skulle ske på projektbasis. När man dock demonstrerade en av demoversionerna av systemet för kundföretagets VD så insåg han att projekt inte alls borde ligga till grund för en faktura utan till exempel skulle flera projekt kunna finnas på samma faktura. Fakturan skulle alltså vara kundbaserad. Detta innebar att man satt, på grund av en kommunikationsmiss, med ett system som byggde på en felaktig grund. Detta innebar att man var tvungen att göra om en del av arbetet. Detta innebar dessutom problem för projektet att hålla tidsramarna.

På grund av den negativa inställningen till förändring var det svårt att motivera projektgruppen till att anpassa systemet till kundens nya krav. Man upplevde att det var kundens problem som hade gett projektet en felaktig grund. Detta ledde till att relationen med kunden och de framtida användarna försämrades.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?

- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Man fick svårt att möta förändringar i de tekniska förutsättningarna. När det visade sig att Internet Explorer inte höll måttet så var man tvungen att byta till PDFer. Detta visade sig bli problematiskt, det fanns en motvilja till att göra om. Det var svårt för vissa i projektet att se att nyttan skulle motsvara arbetsinsatsen. Detta gjorde att resursbehovet ökade.

Man kan konstatera från resultatet av första delen att eftersom man inte funderade kring iterationer eller återkoppling var man tvungen att skriva om stora delar av kodbasen. Den gick ju inte att använda för någonting annat än precis det den var specificerad för. Hade man haft en vettig analys och design och återkopplat till detta regelbundet hade man troligen kunnat se riskerna och bristerna snabbare. Kanske hade man då insett behovet av ett generellare system.

Synen på sekventiellt arbete har förändrats efter utvecklingsprojektets avslut. Återkoppling till analys och design är nu ett viktigt moment vid vidareutveckling av systemet. Man har nog ett mer evolutionsnärt tänkande nu.

Fall ett – ”David”

Projektbeskrivning

David (man, 28 år) var anställd i projektet under den tid projektet löpte. Nu anställd i ett annat mindre företag.

Projektet inleddes med att kunden insåg att den lilla applikation skriven i access i vilken de rapporterade in sin tid, snabb blev begränsad med tiden (ca 3 mån) på grund av ändrade behov och en taskig specifikation. Istället föddes planer på att koppla samman tidrapportering med fakturering. Det skulle skapa ett mer fullständigt affärsstödssystem med integrerad funktionalitet som kunden ansåg sig ha behov av.

Själva utvecklingsarbetet inleddes 2000 med en budget på någonstans kring 400 000 kr. I projektgruppen ingick David samt ytterligare fem personer. Systemet skulle vara webbaserat för att kunna användas av kontor i olika länder.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

Arbetet följde inte någon specifik metod. Istället arbetade vi efter typ best practice. En del av deltagarna hade ju arbetat en del med detta förr och visste hur man skulle lägga upp arbetet. Det var nog inte speciellt medvetet. Arbetet var uppdelat i faser som följde på varandra. En fas gjordes klar innan nästa påbörjades.

I projektgruppen ingick en person som agerade projektledare och som därmed var ansvarig för utvecklingen. ASP och SQL utvecklades i huvudsak av tre av de övriga inkluderade och slutligen så hade man två personer som ansvarade för Javaklienten. David var en av de sistnämnda. Det skulle ju byggas två gränssnitt; webbgränssnittet och sedan även ett administratörsgränssnitt (Javaklienten).

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

David är osäker. Det kändes som ett naturligt sätt att arbeta. Svårt att säga hur arbetet påverkades eftersom man inte vet hur det skulle ha blivit med ett annat upplägg.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Resurser till återkoppling saknades helt i projektet. Detta troligen eftersom man utförde arbetet med analys tidigt i projektet för att skapa en kravspecifikation. Detta skulle sedan inte förändras. När man var klar med kravspecifikationen såg man det som att man hade en god grund för hela arbetet. Denna inställning berodde nog framförallt på okunskap. *"Hela projektet var som en läroprocess. Vi lärde oss otroligt mycket på vägen."* Detta ledde troligen till att man var sämre rustad att möta förändring.

Läroprocessen gjorde att vi som leverantörer inte heller kände att vi kunde fakturera för all tid som vi lade ned. Detta gjorde eventuellt att vissa deltagare i projektet fick en känsla av att det kostar att göra fel. Det blev svårare att möta förändringar som innebar att man var tvungen att göra om arbete som var färdigt.

Projektdeltagarna överlag såg enligt David ett evolutionärt tänkande som väldigt "jobbigt". De tyckte det var mycket bättre att sätta upp en specifikation och en systemdesign för att sedan sätta sig och koda så fort som möjligt. När man kodade vill man helst inte bli avbruten med andra aktiviteter utan istället var det viktigt att man såg snabba resultat i form av nya funktioner hos systemet. De flesta såg dessutom inte processen som en läroprocess, man ville inte riskera kritik genom att till exempel återkoppla och göra om tidigare delar. *"Det sågs som ett riskmoment."*

Användarna var nog med ungefär lika mycket som om man skulle ha arbetat på annat sätt. Kanske att deras medverkan hamnade mest i början och i slutet. I praktiken hade nog användarna ganska lite reellt inflytande över design- och konstruktionsfaserna.

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Slutresultatet var bra. Respondenterna anser att kunden på det hela var nöjd, framförallt med användbarheten. De var inte lika nöjda med att projektet överskred sin tidsplan. Till saken hör också att David och resten av utvecklingsgruppen lade ned mer tid än vad de fick betalt för eftersom den inledande uppskattningen av kostnaderna inte höll. Om detta skulle ha drabbat kunden hade de nog varit betydligt mindre nöjda.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Projektet klarade varken sin tidsplan eller sina resursramar. Båda överskreds även om kunden främst märkte av att tidsplanen överskreds då mycket av de ökade resurserna togs om hand genom att man inte fakturerade för allt.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Systemet, visade det sig, byggde på en specifikation som inte var korrekt. Fakturering skulle helt plötsligt inte byggas på projekt utan på kund. Det gjorde att alla resursramar sprängdes när nu produkten var tvungen att byggas om, vissa delar i stort sett från grunden. Detta var mycket irriterande. Projektledaren var såklart mån om att vara kunden till lags och vi gjorde mycket riktigt om systemet, men det hade nog varit bra om också kunden hade drabbats lite av problemet. Som det var nu blev i och för sig projektet försenat men inte mycket dyrare eftersom vi inte fakturerade för all tid. Det gör ju att kunden inte får en korrekt bild av att det kostar att ändra sig.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Som nämnts tidigare så innebar förändringarna i krav och förutsättningar att resursbehovet ökade en hel del. De ramar som satts upp för resurserna sprack.

Detta beror till viss del på ett tämligen stort praktiskt problem i utvecklingsarbetet. Det visade sig att det fanns ett behov av fakturor på flera sidor. Internet Explorer klarade inte av att rendera så. Istället använde man sig av PDFer. Mycket var tvunget att skrivas om när det var dags att porta fakturorna till PDF. Bristen på något objektorienterat tänkande och återanvändning av kod förvärrade detta. Uppgiftens förändring i och med tekniska problem tvingade fram en omarbetning av delar av pro-

dukten. Detta tog naturligtvis ökade resurser i anspråk. ”*Det skapade problem för projektet att vi inte kunde svara mot förändringar på ett bra sätt. Det är definitivt så att bristen på återkoppling har påverkat projektets resultat negativt.*”

Man ser annorlunda på sekventiellt arbete nu. Det verkar finnas brister med det sättet att arbeta. Samtidigt känns det fortfarande som ett naturligt sätt att lägga upp arbetet. Vet inte riktigt hur det skulle kunna göras annorlunda. Verkar svårt.

Fall två – ”Rikard”

Projektbeskrivning

Rikard är en man som nu är 26 år gammal. Han arbetar fortfarande med systemutveckling, fast nu ofta som formell projektledare.

Leverantör: Ett medelstort utvecklingsföretag. Webbaserade affärslösningar deras specialitet. Företaget hade cirka 12 anställda.

Kund: Ett ganska stort företag som tillverkar specialgjorda ventilationssystem. Patent på framgångsrik teknik.

Tankarna kring projektet uppkom i slutet av 2001 i och med att David träffade ansvariga hos kunden på en privat tillställning. Ett professionellt samarbete inleddes under 2002 då kunden ville ha hjälp med ett utvecklingsprojekt de var på väg att inleda. Det gällde utveckling av en applikation som skulle hjälpa kunden att utföra tekniska beräkningar. Tidigare hade man använt sig av Excel, men det var inte längre tillräckligt.

Projektet hade ingen formell projektledare utan endast en ansvarig från kund. Dock så blev Rikard något av en informell projektledare under resans gång då det ju var denne som hade kontakt med kunden och som gjorde avstämningar. Projektgruppen bestod av fem personer på samma nivå. Två av dessa var Rikard och Caroline. Alla var yngre personer utan utförligare formell utbildning. De gjorde detta lite för att det var kul och spännande och lite för att tjäna pengar. För respondenterna var inte pengarna i fokus.

Beställningen av projektet genomfördes på möten mellan projektgruppen och personer från kund. En grundtanke var att systemet skulle vara generiskt och fungera till även annat än det som var primärt. Detta var en linje som framförallt projektgruppen drev. En offert skrevs och lämnades över vilken innehöll ett fast pris.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

Enligt Rikard var detta ett väldigt problematiskt projekt, av olika anledningar. Arbetet inleddes med att man lade upp en planering som byggde på någon form av sekventiell metod med på varandra följande delar. Det första som gjordes var att ta fram en kravspecifikation i någon form av analysfas. Detta följdes sedan upp genom att man designade delar som till exempel databasen, för att sedan börja skapa funktionalitet. Slutligen så testades alltihop och levererades. Under testningen fick en del arbetas om.

Enligt Rikard slutade analysen i en tämligen svag kravspecifikation, kanske främst beroende på att offerten och beställningen var ganska vag och man endast trodde man hade koll på vad det var man ville göra, vilket kom att förändras.

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

Problemet med upplägget av arbetet var främst att man låstes fast i ett visst tänkande, det blev svårt att möta förändringar. Detta blev ett större problem än i andra projekt eftersom förutsättningarna ändrades redan tidigt i projektet. Nya saker som ”*skulle vara bra om det var med*” kom upp och dessa inkluderades löpande i projektet. Detta gjordes utan någon återkoppling till tidigare arbete. Det hade enligt Rikard varit bättre om dessa saker hade samlats upp och sammanställts till någonting man kunde komma tillbaka till i en framtida vidareutveckling av produkten. Detta för att man då skulle kunna involvera den nya funktionaliteten i projektet även i dess analys- och designrelaterade delar och inte bara i konstruktionen. Så gjorde man hur som helst inte. Det var enligt Rikard svårt att ställa krav på kunden.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Som nämnt tidigare fanns det en svårighet i att möta förändringar. Både i krav och i förutsättningar. På något sätt verkade deltagarna bli mindre benägna att ändra det som var avslutat än det som man fortfarande inte var klar med. Detta skapade problem i och med att man inte påbörjade en fas förrän föregående fas var avslutat. Detta tankesätt genomsyrade även arbetet i faserna. Först gjorde man en sak, sedan nästa osv.

Kunden var endast involverad i den inledande planeringen, alltså i den första av de sekventiella faserna. I de påföljande faserna var kundens medverkan i stort sett obefintlig. Detta borde man enligt Rikard ha ändrat på, det hade man nog tjänat på. Om kunden varit mer involverad hade man kanske inte haft samma problem med förändrade krav i slutet av projektet.

Under projektets gång var Rikard ganska medveten om bristerna i projektarbetet. Dock så jobbade man enligt approachen ”*Vi lägger till detta så fixar vi det också*”. Det var väldigt viktigt för projektdeltagarna, kanske framför allt för Rikard som hade mest kontakt med kunden, att kunden blev nöjd med resultatet.

Inga problem relaterat till graden av dokumentation.

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Kundens reaktion på den slutliga produkten var enligt Rikard på det hela taget dock bra. Han upplever att kunden är nöjd med användbarheten i systemet. Systemet löser, trots vissa brister i funktionaliteten, de problem som var projektets syfte att lösa. Även om projektet i många delar var problematiskt så har Rikard fortfarande en bra relation med kunden som använder systemet i sin organisation på en daglig basis. Rikard är dock själv inte lika nöjde med resultatet. Det hade kunnat bli bättre med bättre förutsättningar.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Tidplanen var på det hela taget helt ok. Det fanns gott om tid till utvecklingsarbetet. Detta gällde naturligtvis även för eventuell evolutionär iterering mellan aktiviteter som man dock ändå inte beagnade sig av.

Vad som är intressant att nämna här är att det fanns bra med i stort sett alla resurser för projektarbetet, framför allt eftersom deltagarna använde även sin fritid till projektet. Detta faktum gör det svårt att avgöra vart resurserna i form av ekonomiska medel gick. Man tog ju inte betalt för all tid som lades ned på arbetet. Man arbetade ju inte bara för pengarna utan även lite för att det var roligt och spännande. ”Hur som helst så var resurser inte begränsande.”

Det innebär ju inte att man höll sig inom givna resursramar, men de sågs inte som en begränsning. Kunden betalade överenskommet pris.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Rikard var involverad i hela projektarbetet i egenskap av projektledare och var enligt honom själv den starkast drivande kraften. Han kände att det var hans ansvar att det skulle bli bra eftersom det var han som höll kontakt med kunden. Alla projektdeltagare var dock involverade, och det var även representant från kunden i viss mån.

Uppgiftens föränderliga karaktär gjorde att det fanns ett stort behov av återkoppling, speciellt eftersom kundens önskemål under projektets gång förändrades nästan lite från dag till dag. I alla fall periodvis. Uppgiften upplevdes av Rikard som väldigt komplex. Han menar att det sekventiella upplägget av arbetet påverkade resultatet negativt, främst med tanke på hur missnöjd kunden var med vissa enskilda delar av funktionaliteten samt med avseende på tidsplanen som man överskred. Kostnaderna hölls inom ramarna främst eftersom man inte fakturerade för all arbetstid.

Detta påverkade dock egentligen inte kundens förväntningar i större utsträckning. Kunden tyckte nog att det var mindre bra att man hade så svårt att möta förändring i projektet, men de flesta av de

problem man upplevde såg aldrig kunden. De lade nog dock märke till att det bitvis gick lite knackigt.

Vad gäller användarmedverkan så tror Rikard att kunden hade velat vara mer involverad. Kanske är deras förändrade krav ett tecken på det? Nya funktioner inkluderades ju dessutom utan att någon sade emot. Det tar Rikard på sitt ansvar eftersom han var den som skötte kontakten med kund. Han borde ha berättat för dem att de inte bara kunde ändra målen för projektet utan att till exempel förändra andra förutsättningar. Även viktigt att analysera de nya behoven.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Under de senare delarna av projektet så försökte man ta upp någon form av återkoppling till tidigare delar även om det fortfarande såklart var bristande. Man ville försöka skapa sig en ny bild av vad det var man skapade. Det visade sig dock mycket svårt att förändra arbetsmodellen under projektets gång.

Vad Rikard upplever var de största nackdelarna med det sekventiella upplägget i projektet var att resursbehovet påverkas negativt av att man inte var rustad att möta förändrade krav från kund eller förändrade förutsättningar. Istället fick man investera mer resurser i form av arbetstid för att det skulle gå ihop i slutändan.

Rikards uppfattning om nackdelarna med sekventiella modeller har förändrats mycket, eller snarare kommit till, under projektets gång. ”*Man är inte dummare än att man lär sig.*”

Fall två – ”Caroline”

Projektbeskrivning

Caroline är nu 27 och arbetar som databasutvecklare på heltid. Hon kom i kontakt med projektet eftersom hon kände Rikard.

Leverantör var ett utvecklingsföretag i vilket hon anställdes för projektet. Kunden var ett företag som tillverkade ventilationssystem.

Projektet inleddes för att utveckla en webbaserad kalkylator. Den skulle hjälpa till med beräkningar både hos kunden och hos deras återförsäljare, som återfinns i ett flertal länder bland annat Tyskland. Därav viktigt att systemet var webbaserat. Tillgänglighet var viktigt.

Totalt var man 5 personer i projektgruppen. Alla yngre personer. Upplevdes som ett spännande projekt. Ett av de första för Caroline.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

Projektet lades upp enligt en tämligen standard modell med på varandra påföljande faser. Ingen specifik process användes. Det upplevdes inte som ett problem, troligen för att vissa uppfattade processer som mest bestående av formalia. Man visste ju vad man skulle göra.

Det visade sig dock att man tydligen inte visste det. Kravspecifikationen blev inte så bra, utan fick arbetas om. Det resulterade i sin tur i många förändringar under arbetet.

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

Upplägget, visade det sig, gjorde att man som projektdeltagare tyckte att det var jobbigt att möta förändring. Man var ju då ofta tvungen att göra om arbete som sågs som avslutat.

Det innebar inte så stora problem under den första fasen, men senare i projektet när man hade en lösning relativt färdigkonstruerad och man hade det man trodde man skulle testa för att sedan leverera uppstod problem. Här trodde Caroline att hennes involvering i projektet var avslutat. ”*Då hade vi skapat det som jag var involverad för att ta hand om.*” Det visade sig dock att kunden inte alls delade uppfattningen om att det var så systemet skulle se ut när det levererades. Man hade nya önskemål om funktionalitet som många verkade uppkomma först när kunden kom i kontakt med en demoversion av systemet. Det var mycket störande. Hade de sagt det tidigare hade man haft bättre möjlighet att möta deras nya krav.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Det fanns klara nackdelar relaterade till möjligheten att möta förändringar i förutsättningarna. Det hade nog dock sett likadant ut om det hade rört sig om andra förändringar, till exempel förändringar i förutsättningar. Det var allmänt ett problem med att möta förändring.

Bristen på iteration upplevdes dock inte som ett stort problem under projektets gång. Detta tror Caroline berodde på att man såg det som väldigt positivt att nå resultat, främst då i form av funktionalitet, vilken man ju gjorde. Problemet var snarare att resultaten inte blev så värdefulla eftersom kraven förändrades löpande. Eventuella skillnader mellan de olika deltagarna berodde enligt Caroline på brist på insikt i projektet, personlighet samt arbetsuppgifter och ansvar.

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Kunden var enligt Caroline nöjd med systemet. Framförallt var kunden enligt Caroline nöjd med användbarheten i systemet. ”*De tyckte att det var lätt att använda.*” Systemet löser även de problem som var anledningen till projektet från första början. Caroline är dock liksom Rikard ovan själv inte lika nöjd med resultatet. Det kommer enligt Caroline bli svårt att utveckla systemet vidare om nya behov skulle uppstå. Hon verkar lite förvånad över att kunden var så nöjd som de blev.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Det fanns i projektet gott om resurser. Man hade varken problem med tid eller andra resurser. ”*Vi hade rätt gott om tid.*” I och för sig genomsyrades arbetet av en mentalitet som innebar att man arbetade till det var klart. Tog det längre tid än beräknat, och det gjorde det, så rapporterade man ändå bara in den överenskomna tiden till kunden. Detta gjorde att kunden fick en felaktig bild av vad projektet egentligen kostade. De visste inte hur mycket resurser som egentligen användes. Kunden var ju nöjda med resursförbrukningen, men Caroline tycker nog inte att man kan säga att man klarade sig inom givna resursramar. Alla problem med förändringarna gjorde ju att allt tog mer tid.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Caroline anser nu att ett fasövergripande arbete är väldigt viktigt. Problemet i projektet var dock att ingen hade dessa kunskaper då och dessutom var de enligt Caroline för snälla. ”*Vi sade inte ifrån fast vi insåg att alla nya förändringar skulle ge ett sämre resultat om de lades till löpande under arbetets gång.*”

Det står enligt Caroline klart att avsaknaden av återkoppling till både projektplanering och till analys/design kraftigt påverkat projektets resultat negativt, framför allt vad gäller systemets användbarhet och möjlighet till vidareutveckling.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Problemet man hade med att möta förändring resulterade helt klart i att projektet krävde mer resurser än vad som hade allokerats inledningsvis.

Dessutom anser Caroline att det vore mycket bättre om kunden involverades mer under hela arbetet eftersom det då skulle ge möjlighet till mer regelbundna avstämningar. Det skulle förhoppningsvis kunna leda till att man kanske kunde undvika reaktioner av den typ som kom från kunden när de såg systemet strax innan leverans. Det var ju inte alls vad de ansåg att man hade kommit överens om. Därav de stora förändringarna och problemet som kostade resurser.

Eftersom man inte hade några problem relaterat till graden av dokumentation fanns det heller inga problem med resurser relaterat till detta.

På det hela taget har Caroline fått en förändrad inställning till sekventiellt arbete. Det känns fortfarande som ett naturligt sätt att lägga upp arbetet på men hon är numer medveten om vikten att löpande gå tillbaka till tidigare faser för att uppdatera det arbetet.

Fall tre – ”Christian”

Projektbeskrivning

Christian var projektets projektledare. Han är 36 år och arbetar idag som CIO på ägarföretaget. Började här strax efter det att verksamheten lagts ner.

Målet var att skapa ett webbaserat spelcommunity, som skulle fungera som gemensam plattform för alla typer av spelintresserade: rollspelare, datorspelare (primärt fokus) och kortspelare.

Möjligheter till chatt, forum, presentationer och framför allt spel (typ Shockwave/Flash. Specialutvecklat för detta community). Spelen kunde organiseras i turneringar eller enskilt. Spelen kostade pengar (virtuella ”coins”) att spela – företagets affärsidé. Turneringar kostade mer men gav möjligheter till att vinna ”coins”.

Start: aug -03, planerad driftsättning vid årsskiftet 03/04. Budget 1,5milj. Systemets verkliga driftsättning var maj -04

Organiserades som dotterbolag till ett stort marknadsföringsföretag, med huvudkontor i Göteborg. Projektets huvudsakliga arbetsplats var också där för där hade man tillgång till både plats och kontorsutrustning. Christian bodde dock i Malmö och veckopendlade.

Projektledaren + 4 anställda togs in i början. Alla relativt unga, dock med högskoleutbildning och arbetslivserfarenhet.

Företaget var självstyrande. Ägarföretaget lade sig inte i särskilt mycket, förutom genom regelbundna möten med projektledaren. Ställde inga direkta krav på produkten.

Augusti 2004 lades projektet ner eftersom man bara hade några hundra medlemmar. Man menade att det inte fanns en marknad för ett community av detta slag och valde att säga upp alla anställda och stoppa fortsatt vidareutveckling av systemet.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

”Nej, det tror jag inte att vi reflekterade över. Vi arbetade på ett sätt som vi kände var naturligt, men någon särskild process har vi inte använt oss av.”

Det var ingenting man diskuterade. Vid projektstarten hade man större frågor att fundera över och diskutera, så att man slapp fundera över det då var bara skönt. I efterhand kanske man kan tänka sig att man borde gjort annorlunda, men det verkade lämpligt då. Christian litade också på de övriga projektdeltagarnas kompetens, och lät dem själva planera mycket av aktiviteterna. *”Jag hade framför allt en säljande funktion; Jag sålde in konceptet dels till ägarna och dels till projektdeltagarna själva, sen fixade de allting själva.”*

Sekventiellt arbetssätt. Man såg fyra moment: Analysera systemets funktionalitet, designa systemet (där man också bestämde sig för utvecklingsplattformar), konstruera det och sedan testa. Man delade upp projektet så här dels för att det verkade naturligt och dels för att man ville vara professionell. Att bara hoppa direkt på att bygga systemet leder bara till ett planlöst kodande.

De två första faserna gick igenom ganska snabbt, medan man lade mer tid på de sista två.

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

”Vi var tvungna att lägga en mängd resurser på att börja utveckla delar som vi senare upptäckte att vi inte kunde eller hade råd att gå vidare med.” Ett exempel är de spelservrar man köpte in och konfigurerade helt i onödan. En mängd andra funktioner i systemet lade man också ner arbetet med.

Dessutom var de specifikationer man skapat tidigt i projektet inte direkt korrekta och passande, men *”jag frågade om de kunde lösa problemen utan att förändra specifikationerna och de svarade ja”*. Man ville inte lägga mer tid än nödvändigt på arbete som inte resulterade i något.

Troligen negativ påverkan.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte under projektets gång.

Om man kan se den spruckna deadlinen som en förändring i en förutsättning, så var just problemet att man fick hoppa över att utveckla vissa delar av systemet.

Man fokuserade inte särskilt mycket på användarna, men det berodde nog inte på upplägget av aktiviteterna. Det berodde nog mer på att man inte förstod bättre än så. Gränssnitten (och logga och grafisk profil) togs till exempel fram av en reklambyrå för att få det snyggt, men det blev ju inte användarvänligt för det.

”Jag kände kanske att man dokumenterade lite litet under projektet, men jag lade mig inte i deras arbete särskilt mycket. Jag var mer samordnare eller nåt.”

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?

- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Det fanns en hel del buggar som verkade göra att medlemmarna blev ganska sura, när det dessutom hela tiden upptäcktes nya.

Det verkade dessutom som att de flesta inte riktigt hängde med i alla de medlemstyper och turneringstyper som fanns, särskilt som de överlappade varandra i viss utsträckning. Detta byggdes dock om senare.

Vissa tyckte nog att det hela var ganska bra, men Christian tror att många var ganska missnöjda med vad de fick för sin medlemsavgift. De grundläggande funktionerna man planerat som till exempel tillgång till en egen presentation, mail till andra medlemmar, gästböcker och dylikt fanns ju, men många lite mer avancerade saker saknades.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Nej. Projektet drog över budget med några hundra tusen och tidsplanen spräcktes med ett par månader. Man fick dessutom ta in en hel del konsulter utifrån för att inte försena projektet allt för mycket.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte under projektets gång.

Många hade nog förväntat sig fler funktioner, som till exempel CounterStrike-serverar, men så blev ju inte fallet.

Antagligen hade det kommit fler besökare och fler medlemmar om man involverat slutanvändarna i högre utsträckning. Å andra sidan hade det tagit längre tid att utveckla systemet då också. De flesta besökare som hittade till communityt tittade bara runt lite för att sedan gå därifrån igen.

Troligtvis ingen större påverkan på graden av dokumentation.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?

- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte

Antagligen har man förlorat en del medlemmar, som förväntat sig fler funktioner, som till exempel Counter-Strike-servrar.

Det har antagligen inte påverkat resursbehovet i någon större utsträckning.

Eftersom man försökte specificera allt och designa allt, fick en del tid läggas ner på att dokumentera delar som sedan inte utvecklades.

Fall tre – ”Nils”

Projektbeskrivning

Nils är 26 år och arbetar idag som programmerare på ett annat medelstort IT-företag.

Projektet startades augusti 2003 och ansågs vara genomfört i maj 2004. Från början togs fyra personer förutom projektledaren in, men det krävdes ett antal ytterligare konsulter under projektets gång. Planen var att vara i drift vid årsskiftet, men blev försenat. De fyra var framför allt tekniker.

Planen var ett community i stil med Lunarstorm, men för en lite äldre målgrupp, med fokus på alla former av spel – Datorspel, rollspel, etc. Tyvärr blev det inte så populärt som de hoppats så ägarföretaget lade ner verksamheten med detta community i slutet av sommaren 2004.

Hur lades arbetet i projektet upp?

- Arbetade ni efter en specifik utvecklingsprocess?
- Fanns en medveten inställning till upplägget av arbetet eller blev det bara så?
- Organiserades arbetet med de olika aktiviteterna sekventiellt över tiden?

”Nej. Vi kände att det var ganska onödigt. Vi hade ju inte med en massa användare att göra och så.” Nils menar att om några användare hade varit inblandade i projektet, hade man antagligen använt sig av en utvecklingsprocess.

Nils menar att medvetenheten kanske fanns att man lade upp arbetet på ett visst sätt, men det var inget man ägnade någon större tanke åt. Att dela upp projektet i mindre delar kändes naturligt för alla

inblandade. Det var inget man diskuterade heller, utan alla verkade ”omedvetet införstådda” med hur man skulle jobba. Projektledaren lade sig inte heller i arbetet särskilt mycket: ”*Det var skönt att kunna fatta mycket av besluten själv, men det märktes ju hur dåligt det var när allt blev mer komplext.*”

Sekventiellt arbetssätt. Först planerades vilka olika delar som communityt skulle innehålla. Efter det planerade man hur allt skulle fungera (Ett UML-diagram) och gränssnittsdesignen köptes in från ett designföretag. Sist byggdes och testades allt innan driftsättning. Driftsättning försenad till 1 maj.

Hur påverkades arbetet i projektet av upplägget av aktiviteterna?

- Var det en positiv eller negativ påverkan?

”*När vi började bygga communityt fattade vi att vår design var kass. Tyvärr hade vi som designade systemet alldeles för dålig kunskap om objektorienterat tänkande. Vi skapade en alldeles för tekniskt komplex design som inte alls överensstämde med det ändå ganska enkla communityt. Till slut blev det helt ohållbart att göra förändringar i systemet. Vi hade inte ens planerat databasen, så den blev bara en enda röra – stor och tung och helt omöjlig att förändra.*”

Nils menar att man inte gick tillbaka och rättade till de dåliga bitarna, trots att man visste att det var dåligt. Isåfall skulle projektet antagligen både bli dyrare och försenat. ”*Det verkade krångligt och dyrt.*” Att ändra specifikationerna kändes onödigt. Man kunde lika gärna skriva lite mer programkod eller fixa saker så att de kunde gå in i databasen.

”*Systemet var så komplext designat att det tog fruktansvärt lång tid att följa systemspecifikationen, även om vi då ansåg att det var värt det eftersom det skulle vara skitenkelt att underhålla och vidareutveckla.*” Det märkte man till exempel när man insåg behovet av säkra Internetbetalningar, då man var tvungen att logga alla transaktioner i databasen. Allt fixades till genom att spara informationen så gott man kunde i någon tabell i databasen och fixa det mesta med mer programkod.

Negativ påverkan.

Vilka nackdelar fanns med upplägget av aktiviteterna?

- Fanns det nackdelar relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte under projektets gång.

Förutsättningarna förändrades inte under projektets gång.

Ja. Projektet började med att bestämma vilka funktioner som skulle finnas, och gick direkt till att designa allt. Gjorde att man inte kunde undersöka vad användarna tyckte förrän efter driftsättning. Alldeles för sent.

Det blev inte särskilt fokuserat på detta. Man ville hellre bygga systemet än hålla på med dokumentation av det.

Hur väl mötte resultatet kundens förväntningar?

- Var kunden nöjd med användbarheten?
- Var kunden nöjd med funktionaliteten?
- Löste systemet kundens problem?

Det beror på vem som är kund. Ägarna brydde sig inte, men användarna verkade tycka att det var ganska krångligt. Massor av olika medlemskapstyper som förändrades hela tiden: man kunde vara flera typer av medlem samtidigt. Inget bra turneringssystem: besökarna förstod inte hur det fungerade. Massa olika typer av turneringstyper som hela tiden kostade olika att vara med i för alla de olika medlemstyperna.

Massor med buggar, som ofta hade allvarliga konsekvenser, som att folk inte kunde köpa nya virtuella pengar eller att pengar drogs från deras kreditkort, men de fick inget för det.

Höll projektet sig inom givna resursramar?

- Klarade projektet sin tidsplan?
- Klarade projektet sin budget?

Nej. Projektet drog över budget, och tidsplanen spräcktes med ett par månader.

Vilka negativa effekter hade eventuella nackdelar på kundens förväntningar?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?
- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte under projektets gång.

Förutsättningarna förändrades inte direkt under projektet.

”Hade vi varit mindre fokuserade på att tillverka specifikationer, och tittat mer på vad besökarna faktiskt ville ha, och haft målgruppen i fokus hela tiden med avstämningar av hur de uppfattade allt, så hade vi nog lyckats mycket bättre.”

Varken ägarföretaget eller slutanvändarna hade ingen någon i utvecklingsprojektet, så det påverkade nog inte i någon större grad.

Vilka negativa effekter hade eventuella nackdelar på resursbehovet?

- ... relaterat till möjlighet att möta förändringar i krav?
- ... relaterat till möjlighet att möta förändringar i förutsättningar?

- ... relaterat till graden av användarmedverkan?
- ... relaterat till graden av dokumentation?

Kraven förändrades inte

Förutsättningarna förändrades inte

Eftersom man inte hade fokuserat alls på användarna av produkten, verkade man inte heller få det genomslag man hoppats på. Man kunde först då se att mycket arbete man gjort fick tänkas igenom ytterligare en gång och ”gå tillbaka till ritbordet”. Till exempel medlemstyper och turneringssystemet. Detta gjorde att man tappade mycket besökare och lite mer än ett halvår efter planerad release (två månader efter faktisk release), hade man fortfarande bara ca 200 betalande medlemmar.

Man skapade ett par specifikationer, men de var inte särskilt ingående. Kravspecen behandlade i stort sett bara vilka funktioner som systemet skulle innehålla, medan systemspecifikationen i princip bara var ett UML-diagram. I och med att man specade alla delar i början, lades en hel del resurser på att utveckla delar som senare skulle väljas bort. Till exempel skulle en Counter-Strike-server sättas upp och integreras med communityn. Man köpte in en server enbart för detta, som konfigurerades och installerades, men när integrationen med communityn skulle börja, visade det sig att det var så mycket arbete att planen lades ner helt.

Samma sak med en realtidschatt.

Kodsammanställning

Nackdel:

N: Typ

Förändringar i systemkraven

Förändringar i förutsättningar

Fokus på dokumentation

Tekniskt fokus och användardistansiering

N-Typ

N-TypSysKrav

N-TypFöruts

N-TypDok

N-TypTekn

N: Påverkan på Customer Satisfaction

Hög och Direkt

Hög och Indirekt

Låg och Direkt

Låg och Indirekt

N-PåvC

N-PåvC-HD

N-PåvC-HI

N-PåvC-LD

N-PåvC-LI

N: Påverkan på Efficiency

Hög och Direkt

Hög och Indirekt

Låg och Direkt

Låg och Indirekt

N-PåvE

N-PåvE-HD

N-PåvE-HI

N-PåvE-LD

N-PåvE-LI

Resultat:

R: Customer Satisfaction

Hög/Låg

R: Efficiency

Hög/Låg

R-CustSat

R-CustSatH/L

R-Eff

R-EffH/L

Sekventiell Modell:

SM: Påverkan på utvecklingsarbetet

Hög/Låg

Positiv/Negativ

SM: Medvetenhet hos utvecklarna

Hög/Låg

SM-Påv

SM-PåvH/L

SM-Påv +/-

SM-Medv

SM-MedvH/L

Exempel på kodning av data

<klipp från fall ett> ”Fakturering skulle ske på projektbasis. När man dock demonstrerade en av demoversionerna av systemet för kundföretagets VD så insåg han att projekt inte alls borde ligga till grund för en faktura utan till exempel skulle flera projekt kunna finnas på samma faktura. Fakturan skulle alltså vara kundbaserad. Detta innebär att man satt, på grund av en kommunikationsmiss, med ett system som byggde på en felaktig grund. Detta innebär att man var tvungen att göra om en del av arbetet. Detta innebär dessutom problem för projektet att hålla tidsramarna.”

Kod: N-TypFöruts samt R-EffL (en nackdel relaterat till förändring i förutsättningar som dessutom innebar en lägre Efficiency)

<klipp från fall två> ”Det visade sig dock att kunden inte alls delade uppfattningen om att det var så systemet skulle se ut när det levererades. Man hade nya önskemål om funktionalitet som många verkade uppkomma först när kunden kom i kontakt med en demoversion av systemet. Det var mycket störande.”

Kod: N-TypTekn samt N-TypSysKrav (nackdelar relaterat till användardistansering som i sin tur ledde till förändringar i krav som man hade problem att hantera).

<klipp från fall tre> ”Nils menar att medvetenheten kanske fanns att man lade upp arbetet på ett visst sätt, men det var inget man ägnade någon större tanke åt. Att dela upp projektet i mindre delar kändes naturligt för alla inblandade. Det var inget man diskuterade heller, utan alla verkade ”omedvetet införstådda” med hur man skulle jobba. Projektledaren lade sig inte heller i arbetet särskilt mycket: ”Det var skönt att kunna fatta mycket av besluten själv, men det märktes ju hur dåligt det var när allt blev mer komplext.”

Kod: SM-MedvL (Låg medvetenhet om det sekventiella arbetssättet)

8. Källförteckning

8.1. Litteratur

- Alter, Steven. *Information Systems – A management perspective*. Boston: Addison-Wesley 1996
- Andersen, Erling S et al. *Målinriktad projektstyrning*. Lund: Studentlitteratur 1994
- Apelkrans, Mats & Åbom, Carita. *OOS/UML*. Lund: Studentlitteratur 2001
- Beck, Kent. *Extreme Programming Explained*. Boston: Addison-Wesley 2001
- Bell, Judith. *Introduktion till forskningsmetodik*. Lund: Studentlitteratur 1995
- Gulliksen, Jan & Göransson, Bengt. *Användarcentrerad systemdesign*. Lund: Studentlitteratur 2002
- Kruchten, Philippe. *The Rational Unified Process: En Introduktion*. Boston: Addison-Wesley 2002
- Löwgren, Jonas. *Human-computer interaction*. Lund: Studentlitteratur 1993
- Miles, Matthew B. & Huberman, Michael. *Qualitative Data Analysis. An Expanded Sourcebook*. Thousand Oaks: Sage Publications 1994
- Patel, Runa & Davidsson, Bo. *Forskningsmetodikens grunder*. Lund: Studentlitteratur 1994
- Preece, Jennifer et al. *Interaction Design: beyond human-computer interaction*. Hoboken: John Wiley & Sons 2002
- Svenning, Conny. *Metodboken - samhällsvetenskaplig metod och metodutveckling*. Eslöv: Lorentz Förlag 1999
- Wallén, Göran. *Vetenskapsteori och forskningsmetodik*. Lund: Studentlitteratur 1996
- Yin, Robert K. *Case Study Research: Design and Methods*. Thousand Oaks: Sage Publications 2003

8.2. Artiklar

- Boehm, Barry W. "A Spiral Model of Software Development and Enhancement", *Computer*, nr 5 1988, s 61

Cavaye, Angéle. "User participation in system development revisited", *Information & Management*, nr 28 1995, s. 311

Cotton, Todd. "Evolutionary fusion: A customer-oriented incremental life cycle for fusion", *Hewlett-Packard Journal*, nr 4 1996, s 25

Dvir, Dov & Lechler, Thomas. "Plans are nothing, changing plans is everything: the impact of changes on project success", *Research Policy*, nr 1 2004, s 1

Hull, M.E.C et al. "Software development processes – an assessment", *Information and Software Technology*, nr 44 2002, s 1

Lipovetsky, Stan et al. "The relative importance of project success dimensions", *R & D Management*, nr 2 1997, s 97

Rollo, Anthony L. "Jackson System Development", Introduction to Software Design Methodologies, IEE Colloquium on, 1992, s 3/1-313

Royce, Winston W. "Managing the development of large software systems: concepts and techniques", *International Conference on Software Engineering*, 1987, s 328

Russo, Nancy & Stolterman, Erik. "Exploring the assumptions underlying information systems methodologies", *Information Technology & People*, nr 4 2000, s 313

Sommerville, Ian. "Software Process Models", *ACM Computing Surveys*, nr 1 1996, s 269

Woods, Peter et al. "A research team in ethnography", *International Journal of Qualitative Studies in Education*, nr 1 2000, s 85

Zeffane, Rachid et al. "Does user involvement during information systems development improve data quality?", *Human Systems Management*, nr 2 1998, s. 115

8.3. Övrigt material

Bendelius, Niklas & Jönsson, Mattias & Olsson, Ols Jonas Petter. *En utvecklarens mardröm – Viktiga val i projektet InKund*. Lund: Lunds Universitet, Institutionen för Informatik 2003