

# **LEDER ANVÄNDNINGEN AV RATIONAL UNIFIED PROCESS TILL EN FÖRBÄTTRAD SYSTEMUTVECKLINGSPROCESS?**

Författare: Mattias Johannesson  
Handledare: Maria Bergenstjerna

---

## Abstrakt

I dagens växande IT-samhälle med ständig förändring och överflöd av information, finns det idag ett problem i många organisationer och det är att de inte använder sig av någon gemensam systemutvecklingsmetod, utan de använder sig av flera olika eller egenhändigt skapade metoder. Det har funnits otaliga systemutvecklingsmetoder som använts de senaste decennierna, men behoven har på senare tid förändrats och dessa metoder stödjer i många fall inte längre processen på ett tillfredsställande sätt. Det har på senare år vuxit fram kritik mot det traditionella sättet att utveckla system. Kritiken består bland annat av problem med att möta behov från både ledning och kunder, brist på kontroll, samt problem med dokumentation. Ivar Jacobson, Grady Bootch och James Rumbaugh har skapat en ny metod för att utveckla system nämligen att använda sig av RUP (Rational Unified Process). Syftet med denna studie var att utreda om användningen av RUP leder till en förbättrad systemutvecklingsprocess. Jag har genom intervjuer samlat in information och tolkat denna utifrån min analysmodell. Studien visar att RUP leder till en förbättrad systemutvecklingsprocess genom att den hanterar de flesta av problemen som fanns med de tidigare metoderna.

## **FÖRORD**

Hösten 2001 läste jag kursen Informationssystemmiljöer, och i denna kurs ingick en 5 poänguppsats där jag skrev om prototyputveckling. I och med denna uppsats kom jag i kontakt med flera olika metoder för systemutveckling och blev intresserad av RUP.

Jag vill tacka min handledare Maria Bergenstjerna som hjälpt mig när jag kört fast. Jag vill även tacka respondenterna som ställt upp på intervjuerna, och mina vänner som hjälpt mig hitta dessa kunniga personer.

Göteborg januari 2003

Mattias Johannesson

## INNEHÅLLSFÖRTECKNING

<b>1 INTRODUKTION</b> .....	<b>5</b>
<b>1.1 BAKGRUND</b> .....	<b>5</b>
<b>1.2 SYFTE</b> .....	<b>5</b>
<b>1.3 FRÅGESTÄLLNING</b> .....	<b>5</b>
<b>1.4 AVGRÄNSNING</b> .....	<b>6</b>
<b>1.5 DISPOSITION</b> .....	<b>6</b>
<b>2 VETENSKAPLIG METOD</b> .....	<b>7</b>
<b>2.1 KUNSKAPSSYN</b> .....	<b>7</b>
2.1.1 Empirism .....	7
2.1.2 Positivism .....	7
2.1.3 Hermeneutik .....	8
<b>2.2 OLIKA ANGREPPSSÄTT</b> .....	<b>8</b>
2.2.1 Induktion.....	8
2.2.2 Hypotetisk – deduktiva.....	9
2.2.3 Abduktion.....	9
<b>2.3 OLIKA TYPER AV UNDERSÖKNINGAR</b> .....	<b>9</b>
2.3.1 Klassificering med utgångspunkt från datakällor och metoder för datainsamling .....	9
2.3.2 Klassificering med utgångspunkt från undersökningens syfte.....	9
2.3.3 Klassificering med utgångspunkt från undersökningens typ av data.....	10
<b>2.4 DATAINSAMLING</b> .....	<b>11</b>
<b>2.5 VALIDITET OCH RELIABILITET</b> .....	<b>11</b>
<b>2.6 SAMMANFATTNING AV MINA VAL</b> .....	<b>12</b>
<b>3 TEORI</b> .....	<b>13</b>
<b>3.1 PROCESS</b> .....	<b>13</b>
<b>3.2 FRÅN SYSTEMS DEVELOPMENT LIFE CYCLE TILL SPIRALMODELLEN</b> .....	<b>13</b>
3.2.1 The system development life cycle .....	13
3.2.2 Potentiella svagheter med the traditional life cycle: .....	15
<b>3.3 SPIRALMODELLEN</b> .....	<b>17</b>
<b>4 VAD ÄR RUP?</b> .....	<b>18</b>
<b>4.1 BEST PRACTICES</b> .....	<b>18</b>
4.1.1 Arbeta iterativt.....	18
4.1.2 Hantera kraven.....	19
4.1.3 Utveckla komponentbaserat.....	19
4.1.4 Modellera visuellt.....	19
4.1.5 Verifiera kvalitén kontinuerligt.....	20
4.1.6 Hantera förändringar .....	20
<b>4.2 RUP:S UPPBYGGNAD</b> .....	<b>21</b>
4.2.1 Den horisontella dimensionen .....	21
4.2.2 Den vertikala dimensionen .....	24
4.2.3 Processens statiska struktur: .....	26
<b>5 PROCESSFÖRBÄTTRING</b> .....	<b>28</b>
<b>5.1 SOMMERVILLE</b> .....	<b>28</b>
<b>5.2 HÄGERFORS</b> .....	<b>28</b>
<b>5.3 CHECKLAND &amp; SHOLES</b> .....	<b>29</b>
<b>5.4 ANALYSMODELLEN</b> .....	<b>30</b>
<b>6 EMPIRI</b> .....	<b>31</b>
<b>7 RESULTAT OCH ANALYS</b> .....	<b>32</b>
<b>7.1 SOMMERVILLE</b> .....	<b>32</b>
7.1.1 Understandability – förstålighet.....	32
7.1.2 Visualibility – synlighet .....	34
7.1.3 Supportability – stödjande.....	35

7.1.4 Acceptability – grad av acceptans.....	36
7.1.5 Reliability – reliabilitet.....	38
7.1.6 Robustness – robusthet.....	39
7.1.7 Maintainability – varaktighet.....	40
7.1.8 Rapidity – snabbhet.....	42
<b>7.2 HÄGERFORS</b> .....	43
<b>7.3 CHECKLAND &amp; SCHOLES</b> .....	45
7.3.1 Efficiency.....	45
7.3.2 Efficiency.....	45
7.3.3 Effectivness.....	46
7.3.4 Ethicality.....	46
7.3.5 Elegance.....	47
<b>7.4 SAMMANFATTNING</b> .....	47
<b>8 DISKUSSION</b> .....	<b>49</b>
<b>9 FRAMTIDA FORSKNING</b> .....	<b>49</b>
<b>10 REFERENSER</b> .....	<b>50</b>
<b>APPENDIX MED INTERVJUERNA</b> .....	<b>51</b>
<b>A.1 SOMMERVILLE</b> .....	51
A.1.1 Understandability – förstålighet.....	51
A.1.2 Visualibility – synlighet.....	54
A.1.3 Supportability – stödjande.....	56
A.1.4 Acceptability – pålitlig.....	58
A.1.5 Reliability – reliabilitet.....	60
A.1.6 Robustness – robusthet.....	61
A.1.7 Maintainability – varaktighet.....	62
A.1.8 Rapidity – snabbhet.....	64
<b>A.2 HÄGERFORS</b> .....	65
<b>A.3 CHECKLAND&amp;SCHOLES</b> .....	67
A.3.1 Efficiency.....	67
A.3.2 Efficiency.....	68
A.3.3 Effectivness.....	68
A.3.4 Ethicality.....	69
A.3.5 Elegance.....	69

# 1 INTRODUKTION

*I detta kapitel beskrivs arbetets bakgrund, syfte, frågeställning, avgränsning och en disposition över arbetet.*

## 1.1 Bakgrund

Det finns ett problem i många organisationer och det är att de inte använder sig av någon gemensam utvecklingsmetod, utan de använder sig av flera olika eller egenhändigt skapade metoder, detta kan leda till ostrukturerade och okontrollerade projekt.

Det har funnits otaliga systemutvecklingsmetoder som används de senaste decennierna, men behoven har förändrats på senare tid och dessa metoder stödjer i många fall inte längre processen på ett tillfredsställande sätt. Processen har i många fall en systemutvecklingsmetod som styrmedel för att effektivisera arbetssättet, och kunna kontrollera tidsplanerna. För att detta ska kunna ske måste tillvägagångssättet dokumenteras för att kunna återanvändas och upprepas i kommande projekt. (Avison & Fitzgerald, 1997)

Det har på senare år vuxit fram kritik mot det traditionella sättet att utveckla system. Problem med att kunna möta behov från både ledning och kunder. Brist på kontroll, problem med dokumentation osv.. Ivar Jacobson, Grady Bootch och James Rumbaugh har skapat en ny metod för att utveckla system nämligen att använda sig av RUP( Rational unified process), i denna systemutvecklingsmetod sker utvecklingen iterativt till skillnad gentemot det traditionella sekventiella tillvägagångssättet. Som jag nämnt tidigare finns det företag som inte har någon gemensam systemutvecklingsmetod, och det finns behov av en standard. Enligt Rational möter de kundens krav gällande budget och tid, men leder detta till att processen förbättras? (Kruchten, 2000)

När man arbetar med systemutveckling är det vanligt att man arbetar i projektform, och fokuseringen ligger i många fall på att försäkra sig om att slutprodukten, dvs. mjukvaran håller hög kvalitet. Att utveckla en produkt innebär att gå igenom en process, alltså själva vägen till målet. Går det att mäta och förbättra denna process?

## 1.2 Syfte

Syftet med denna uppsats är initialt att öka förståelsen av systemutvecklingsmetoden RUP. Jag har undersökt hur tillämpningen av RUP ser ut i olika organisationen, med olika tid av användning. Det primära syftet med uppsatsen är att undersöka om systemutvecklingsprocessen förbättras genom användningen av RUP.

## 1.3 Frågeställning

Min huvudfråga i uppsatsen är:

- Leder användningen av RUP till en förbättrad systemutvecklingsprocess?

För att kunna besvara huvudfrågan kommer följande delfrågor utredas:

- Vilka är fördelarna/nackdelarna med denna metod?
- Är kritiken befogad mot det traditionella tillvägagångssättet?
- Är RUP lösningen på denna kritik?

## **1.4 Avgränsning**

Uppsatsen behandlar inte Rational's olika verktyg, utan jag har avgränsat mig till att analysera metoden Rational unified process. Inte heller standardnotationen unified modelling language UML, som verktygen använder sig av tas upp. Förbättring beaktas endast angående processen, ej av produkten.

## **1.5 Disposition**

### **Kapitel 1 - Introduktion**

Beskriver uppsatsens bakgrund, syfte, frågeställning, och avgränsning.

### **Kapitel 2 - Vetenskapsmetod**

Beskriver kunskapssyn, olika angreppssätt, och olika typer av undersökningar.

### **Kapitel 3 - Teori - bakgrund**

Teoriavsnittet börjar med en beskrivning av vad en process är och sedan följer en beskrivning av det traditionella sättet att utveckla system, dess svagheter och dess efterföljare spiralmodellen.

### **Kapitel 4 - Definition av RUP**

En genomgång av vad RUP är, samt dess uppbyggnad..

### **Kapitel 5 – Process förbättring**

Vad som karakteriserar en framgångsrik process, samt min analysmodell som används till empirin och analysen beskrivs

### **Kapitel 6 - Empiri**

En beskrivning av respondenterna

### **Kapitel 7 – Resultat och Analys**

Resultat och analys av respondenternas svar relaterat till analysmodellen.

### **Kapitel 8 - Diskussion**

### **Kapitel 9 - Framtida forskning**

### **Kapitel 10 – Referenser**

## **APPENDIX**

## 2 VETENSKAPLIG METOD

*I detta kapitel beskrivs till en början olika vetenskapliga förhållningssätt till kunskap, för att sedan beskriva olika angreppssätt, och sist en redovisning av olika typer av undersökningar.*

### 2.1 Kunskapssyn

Alla människor har olika värderingar och utifrån dessa tolkar man händelser och intryck från verkligheten. Med en värdering avses en föreställning om ett önskat tillstånd eller mål. En person kan ha en uppsättning värderingar av den egna situationen, om sociala relationer eller politiska förhållanden. De värderingar som finns genomsyrar mer eller mindre hela samhället där en individ befinner sig. Vidare skapas en personlig referensram som påverkar beteende och även egna arbeten och utredningar. Utifrån denna kontext kan inte heller forskningen förhålla sig helt objektiv, i och med att man påverkas av de värderingar som medvetet bär med objektivitet, synen på objektivitet och subjektivitet skiljer sig mellan olika forskningsinriktningar.

En positivistisk forskare har stora krav på sig att ha ett objektivt förhållningssätt. Till skillnad från den positivistiska, är den hermeneutiska forskningen präglad av subjektivitet. Dessa forskare använder sig medvetet av värderingar vid genomförande av forskningsprocesser. De kunskaper och erfarenheter som forskaren har utgör hans förförståelse. (Wiedersheim-Paul & Eriksson, 1993)

#### 2.1.1 Empirism

Empirisk kunskap kännetecknas av att kunskaperna grundas på observationer av verkligheten. Empiri kan ungefär översättas med erfarenhet. Empirism betyder erfarenhetsinriktning och empirismen hävdar att all kunskap bygger på observationer av verkligheten. Empirismen preciserar att man ska studera fenomenen sådana de uppträder och att en förståelse av samhället måste ta sin utgångspunkt i detaljstudier. De flesta vetenskaper är empiriska. (Andersson, 1994)

#### 2.1.2 Positivism

Positivism har sina rötter i en empirisk/naturvetenskaplig tradition. Den franska sociologen Auguste Comte gav namnet åt positivismen. Han var verksam på 1800-talet. Han menade att det går att generera kunskap som var positiv (säker) och utvecklande för mänskligheten. Kunskapen man sökte skulle vara verklig och tillgänglig för våra sinnen och förnuft. För att bli positiv, skulle kunskapen vara nyttig och kunna förbättra samhället, säker genom att den byggde på iakttagelser som var logiskt prövbara, exakt genom att komplexa företeelser reducerades till sina enkla beståndsdelar och organiserad genom att den formulerades som lagbundenheter. (Patel & Davidsson, 1993)

Grundproblemet gällde vad som var utmärkande för vetenskaplig kunskap. Man formulerade den kända "verifierbarhetstesens"; att en vetenskaplig sats är meningsfull bara om den kan verifieras empiriskt, eller om något annorlunda men enklare uttryckt; ett påstående är sant om det överensstämmer med verkligheten. Allt som inte kunde prövas empiriskt, som känslor och värderingar, religiösa och politiska uttalanden osv.. hörde inte till den vetenskapliga sfären. Det finns olika drag i dagens positivism bl a. tilltron till vetenskaplig rationalitet, kunskapen

skall vara empirisk prövbar, uppskattningsbar och bedömningsbar, skall ersättas med mätningar. Det finns inom vetenskaplig metod regler och kunskapskrav. Metoderna ska ge tillförlitlig kunskap. Detta kan preciseras i mätkrav som validitet och reliabilitet. Förklaringar ska kunna anges i termer av orsak - verkan och kunskapen i lagbundenheter. Forskaren ska vara objektiv och inte låta sig påverkas av utomvetenskapliga värderingar.

Den traditionella positivistiska uppfattningen innebär att forskningen både kan och ska stå utanför etik och moral. (Andersson, 1994)

### *2.1.3 Hermeneutik*

Det råder en avgörande skillnad mellan fysiska och sociala fenomen. Alla fenomen tolkas och tyds av människan, fenomenen får betydelser, innebörd och mening. Att forska om sociala fenomen handlar bland annat om att förstå betydelser.

Målet för forskningen är att uttolka och att förstå. Att studera ett fenomen och försöka förstå det, innebär med nödvändighet en uttolkning. Det första steget är problematisering - att se något som något.

Enskilda fenomen kan endast förstås genom att förstå det speciella sammanhang i vilket det ingår, man behöver en helhetssyn. Det styrs också av vem som gör tolkningen samt ur vilket perspektiv och med vilket språk den görs.

Det går inte alltid att skilja mellan faktaomdömen och värdeomdömen. Känslor kan förmedla kunskaper som man inte kan nå genom förnuftet. Forskaren bör med inlevelse och engagemang söka tränga in i, delta i, det fenomen som skall studeras. Det är omöjligt, och inte alltid önskvärt att bedriva helt opartisk forskning.

Syftet med forskning är i hög grad att förstå men kan också gå ut på att åstadkomma förändring och studera vad som då sker. Personliga erfarenheter är ofta nödvändiga förutsättningar för att uppnå vetenskaplig kunskap.

Den tydligaste skillnaden gentemot positivismen är att man inte håller forskarens värderingar utanför studien, samt att man inom hermeneutiken är bunden till tid och rum. (Lundahl & Skärvad, 1992)

## **2.2 Olika angreppssätt**

Val av forskningsansats berör i hög grad uppfattningar om relationerna mellan teori och empiri. Man brukar tala om två ”metodansatser”: den induktiva och den hypotetisk-deduktiva metoden, men under senare tid har man börjat tala om ett tredje sätt, en medelväg som kallas abduktion.

### *2.2.1 Induktion*

Induktion innebär att man drar allmänna, generella, slutsatser utifrån empirisk fakta. Induktion förutsätter alltså kvantifiering. Induktion utgår ifrån att man studerat en stor mängd enskilda fall och hävdar att ett funnet samband i alla dessa enskilda fall är generellt giltigt. Man kan mer konkret säga att man utgår ifrån empirin och skapar en teori. En induktiv slutledning är svårt att fullt ut lita på då empiriskt material sällan består av en fullständig uppräkningslista.



### 2.2.2 Hypotetisk – deduktiva

I den *hypotetisk - deduktiva* metoden ställer man upp hypoteser och gissningar, som premisser. Sedan gör man en deduktiv slutledning, och till slut undersöker man om premisserna stämmer med verkligheten. På det sättet använder man sig både av empiri och logik. Deduktion innebär att man drar en logisk slutsats som betraktas som giltig om den är logiskt sammanhängande. Däremot behöver den inte nödvändigtvis vara sann i den meningen att den överensstämmer med verkligheten.

### 2.2.3 Abduktion

Abduktion är ett sätt att dra slutsatser om vad som är orsak till eller har föregått en observation. I den hypotetisk - deduktiva ansatsen undersöks samband genom att man varierar olika tänkbara orsaksfaktorer och sedan undersöker vad effekterna blir. Abduktion står för oväntad kunskap. Det är en kombination av både induktion och deduktion. Abduktion innebär att tidigare kända mönster avger det fenomen som undersöks ses på ett nytt sätt, ur ett nytt perspektiv. Detta leder till att en abduktiv slutledning avleder från ett väntat mönster. Abduktion utgår från empirisk fakta men avvisar inte teoretiska fakta. Denna vetenskapsmetod kallas även för den gyllene medelvägen.

Abduktion gäller en delvis omvänd situation, fall där man står inför en effekt och söker orsaksfaktorer utan att direkt kunna manipulera dessa. Man utgår ifrån sannolika samband, drar slutsatser genom uteslutningar av olika faktorer, kompletterar med tester osv.. Abduktion är ingen metodik som kan användas schematiskt utan kräver ingående erfarenheter av det område frågorna gäller, erfarenhet av liknande fall m.m. Slutsatserna gäller inte strikt logiskt utan måste provas vidare genom praktiska försök. (Thurén, 1991), (Wallén, 1996)

## 2.3 Olika typer av undersökningar

### 2.3.1 Klassificering med utgångspunkt från datakällor och metoder för datainsamling

Datakällor kan delas in i ”paper and people”. Informationen kan inhämtas från dokument av olika slag (”paper”) exempelvis böcker, artiklar, rapporter osv.. Å andra sidan kan data inhämtas från människor (people). Informationen inhämtas då oftast genom intervjuer, enkäter och observationer. Datakällor kan vidare indelas i primärdata dvs. av utredaren själv insamlat material, och sekundärdata dvs. av andra insamlat material. (Lundahl & Skärvad, 1991)

### 2.3.2 Klassificering med utgångspunkt från undersökningens syfte

#### *Explorativa undersökningar*

Med explorativa undersökningar avses undersökningar som syftar till problemformulering och problemprecisering, ofta uttryckt i hypoteser, samt att orientera utredaren i utredningens frågeställning och om vad som är känt i ämnet redan tidigare. Att komma fram till en preciserad och sammanhängande undersökningsplan med frågeställningar, syfte, metoder för datainsamling, analys etc. Vanligtvis använder utredaren expertintervjuer, litteraturgenomgång och fallstudier

### *Förklarande undersökningar*

Hypotesprövande eller förklarande undersökningar har som syfte att studera kausala samband. Utifrån redan existerande teorier och den kunskap forskaren har ställer han upp hypoteser eller påståenden som han sedan vill testa mot verkligheten. Den traditionella metoden i dessa sammanhang är experiment.

### *2.3.3 Klassificering med utgångspunkt från undersökningens typ av data*

Man brukar skilja mellan två metodologiska angreppssätt med utgångspunkt från vilken typ av information som ska studeras; - hårddata eller mjukdata, och då talar man om kvantitativa eller kvalitativa metoder. Den viktigaste skillnaden mellan dessa ligger i hur man hanterar siffror och statistik.

#### *Kvantitativ metod*

I de kvantitativa metoderna omvandlar forskaren informationen till siffror och mängder och försöker sedan analysera materialet med hjälp av statistiska modeller. Kvantitativ forskning som syftar till att beskriva och förklara det som undersöks, har ett utifrånperspektiv.

Forskaren har som uppgift att neutralisera det subjektiva inslagen i hämtad information och studera denna på ett objektivt sätt. Det bör finnas en entydig beskrivning av fenomenet, att ha en begreppsapparat, en modell som visar samband och mätbarhet. Datainsamling sker genom survey eller enkätundersökning. (Thurén, 1991)

#### *Kvalitativ metod*

Inom kvalitativa metoder försöker forskaren sammanställa informationen för att skaffa sig en helhetssyn. Det är sedan dennes uppfattning och tolkning av materialet som är det avgörande. Det bör kanske poängteras att skillnaden mellan metoderna inte är absolut. Båda utgör någon form av redskap eller verktyg som i större eller mindre utsträckning använder sig av diverse metodiska principer. Det finns heller inte något konkurrensförhållande mellan dem, utan de kan i många fall komplettera varandras styrkor och svagheter. Den mest påtagliga likheten mellan metoderna är att de båda syftar till att öka förståelsen av det samhälle vi lever i, och dess människor, samt deras inbördes påverkan.

Inom kvalitativa metoder är det forskarens uppfattning eller tolkning av information som står i centrum. Kvalitativa metoder har sin styrka i att de visar totalsituationer. En sådan helhetsbild möjliggör en ökad förståelse för sociala processer och sammanhang.

Kvalitativa studier präglas av flexibilitet på två sätt. För det första måste man kunna ändra på uppläggningsen under själva genomförandet av undersökningen. Om man under undersökningens gång upptäcker att vissa frågeformuleringar glömts bort eller formulerats fel, rättar man till detta. För det andra är uppläggningsen flexibel i förhållande till det sätt på vilket man närmar sig de olika undersökningsenheterna. Det gäller både vilka frågor som tas upp och i vilken ordningsföljd.

#### *Konkreta kännetecken för kvalitativ metod är:*

- ger riklig information om få undersökningsenheter. Man går in på djupet
- osystematiska och ostrukturerade observationer t ex.. en intervjumall utan fasta frågor eller svarsalternativ
- man intresserar sig för det säregna, det unika eller det eventuellt avvikande
- insamlingen av informationen sker under förutsättningar som ligger nära den verklighet

man vill undersöka

- man intresserar sig för sammanhang och strukturer
- forskaren observerar fenomenet inifrån. Han kan även delta som aktör jag - du - relation mellan forskaren och den undersökte

Syftet med kvalitativa intervjuer är att öka informationsvärdet och skapa en grund för djupare och mer fullständiga uppfattningar om det fenomen man studerar. I en kvalitativ intervjuprocess kommer man ofta in på saker som är av mycket privat karaktär. (Holme & Solvang, 1997)

## 2.4 Datainsamling

Datakällor brukar delas upp i två olika kategorier; sekundär- och primärdata. Primärdata är det som forskaren själv har samlat in, medan sekundärdata är data som redan finns tillgänglig.

Med primärdata avser man information som samlats för just denna undersökning. Primärdata för marknadsundersökningar erhålls ofta av enskilda eller grupper av respondenter. Tekniker för insamling av data är exempelvis intervjuer och enkäter.

Sekundärdata avser man data som inte är insamlad för just denna undersökning. Det kan innebära att man fått sitt material ur facktidningar, statistik eller tidigare undersökningar. Problemet med sekundärdata är att det är insamlat för annat ändamål och därför inte alltid ger svar på de frågor forskaren har för sin undersökning. Det kan många gånger vara bra att använda sekundärdata som utgångspunkt för ett vidare fördjupande arbete, vilket kompletteras med primärdata. Det är viktigt att forskaren är kritisk till det material som används och sällar bort fakta som inte uppfyller krav på tillförlitlighet. (Lundahl & Skärvard, 1992) (Holme & Solvang, 1997)

## 2.5 Validitet och reliabilitet

För att kunna skriva ett bra arbete är det väldigt viktigt att den information som samlats in är pålitlig (reliabilitet) och giltig (validitet). Validitet och reliabilitet står i ett visst förhållande till varandra som gör att vi inte bara kan koncentrera oss på det ena och låta bli det andra. Validitet innebär att man verkligen undersöker det som var avsikten att undersöka. Reliabilitet betyder att vi gör undersökningen på ett tillförlitligt sätt.

Reliabiliteten är en grundläggande förutsättning för att veta om informationen som samlats in representerar det som har mätts dvs. om informationen är valid. Validitet innebär att man verkligen har undersökt det som varit till avsikt att undersöka och ingenting annat. Det är inte tillräckligt att ha pålitlig information. Om man råkar mäta något helt annat än det som var för avsikt, kan den informationen vara hur pålitlig som helst, men den kan då inte användas i rätt syfte. En förutsättning för detta är också att man har giltig information. (Thurén, 1991)

En förutsättning för att forskningsresultatet ska kunna förklara faktiska förhållanden är att de fakta man observerat är pålitliga dvs. att man kommer fram till ungefär samma resultat när man mäter samma fenomen på olika sätt. Man kan skilja mellan intersubjektiv reliabilitet och intrasubjektiv reliabilitet. Med intersubjektiv reliabilitet menar man graden av överensstämmelse mellan olika forskares mätningar av samma fenomen. Med intrasubjektiv

reliabilitet menar man graden av överensstämmelse mellan samma forskares olika mätningar av samma fenomen.

Även om undersökningen har hög reliabilitet behöver den inte ha validitet. Man säger att ett resultat visar hög validitet när man mäter hela det fenomen som man avsett att mäta och inget annat. Vid intern validitet faller överensstämmelsen mellan innehållet på den nominella och den operationella nivån. Extern validitet gäller ett resultats överensstämmelse med verkligheten dvs. om det är sant eller falskt. Den externa validiteten kan bara konstateras genom en empirisk prövning. (Andersson, 1994)

## **2.6 Sammanfattning av mina val**

### **Kunskapssyn**

Jag har använt mig av ett hermeneutiskt förhållningssätt för att öka min grundläggande förståelse inom det område jag valt att studera. Jag har tolkat och analyserat den information jag erhållit genom uppsatsen.

### **Angreppssätt**

Jag har använt mig av en abduktiv ansats då jag alltefter arbetes gång växlat mellan befintliga teorier och empiri.

### **Olika typer av undersökningar**

Jag har använt mig av en kvalitativ undersökningsmetod eftersom jag försökt skaffa mig djupare förståelse inom ett specifikt område. Jag har sammanställt, tolkat, och analyserat informationen för att försöka få en helhetssyn.

### **3 TEORI**

*I detta avsnitt ges till en början en beskrivning av vad en process är, vidare beskrivs det traditionella sättet att utveckla system, och dess problem. Sist en beskrivning av Spiralmodellen.*

#### **3.1 Process**

Begreppet process kommer från latinets "processus" och "procedere" som ungefär betyder "framåtskridande" eller "gå framåt". Ordet process har kommit att beteckna olika typer av förlopp, något som utvecklas i tiden, och ofta långsamt. Bergman & Klefsjö definierar en process som "en process är ett nätverk av aktiviteter som upprepas i tiden och vars syfte är att skapa värde åt någon extern eller intern kund".

En sådan definition kan dock ge en mekanisk syn, där processer uppfattas som maskiner eller löpande band. I verkligheten handlar processer till stor del om koordination mellan människor, det vill säga överenskommelsen mellan individer som samverkar och om dessa individers kompetens. (Bergman & Klefsjö, 2001)

En process består av en serie av aktiviteter med preciserad början och slut, och som med hjälp av en organisations resurser, återkommande förädlar ett mätbart objekt från en leverantör till ett på förhand bestämt mätbart resultat till en kund. (Karlsson & Söderstedt, 1997)

Gå igenom en serie av förutsägande steg – en vägkarta som hjälper till att skapa en slutprodukt med hög kvalitet i tid. Viktigt eftersom processen ger stabilitet, kontroll till organisationen. (Pressman, 2001)

Mjukvaruprocess är en uppsättning av aktiviteter och associerade resultat vilka leder till produktion av en mjukvaruprodukt. (Sommerville, 2001)

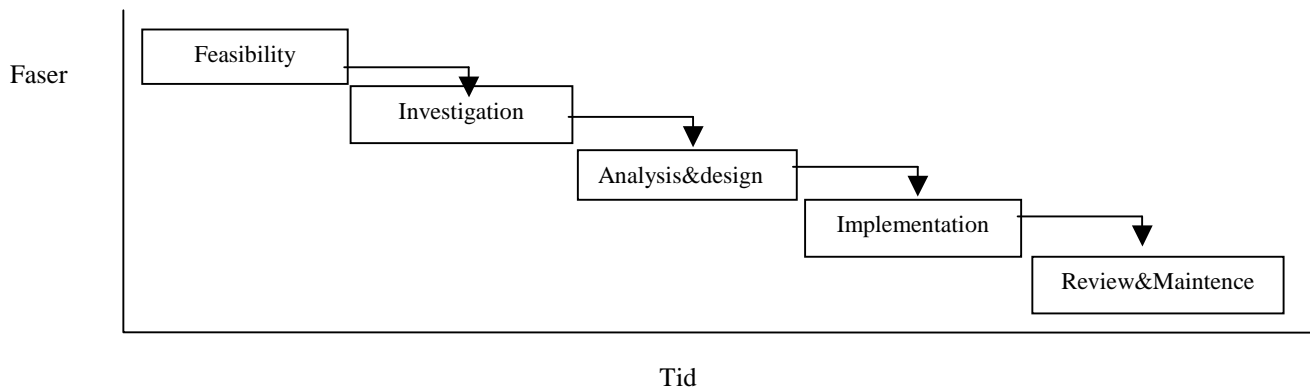
#### **3.2 Från systems development life cycle till spiralmodellen.**

System development life cycle växte fram som ett svar på de problem som fanns inom systemutvecklingen under 50 och 60-talet, då systemutvecklingen ofta förlitade sig på en programmerare. Denna person var inte alltid insatt i användarnas behov och önskemål, utan gjorde en lösning som ansågs som passande. Dokumentationen var så gott som obefintlig, så när ett problem uppstod så fick i många fall utvecklarna börja om ifrån början. Det var i slutet på 60 och början på 70-talet som denna metod växte fram. Följande beskrivning av the system development life cycle baseras på Avison och Fitzgerald. (Avison & Fitzgerald, 1997)

##### *3.2.1 The system development life cycle*

Denna traditionella metod har använts i årtal för att t ex.. skapa grundläggande applikationer såsom löneutbetalning, bokföring osv.. Arbetet som ska utföras delas in i väldefinierade steg, och arbetsprocessen är kontrollerat av formella projektledningsmetoder. Arbetsgången är linjär dvs. att när ett steg är avklarat så fortsätter arbetet i nästa steg utan någon iteration av de tidigare stegen. Den är mest effektiv i skapandet av system där kraven är väl införstådda och kan bli väl specificerade, före någon detaljdesign eller kodning av programmet börjat.

System development life cycle har följande steg:



Figur 1. System development life cycle (Avison & Fitzgerald, 1997)

### Feasibility / Förstudie

En undersökning görs på det aktuella systemet, de krav det var tänkt att uppfylla, problem att uppfylla dessa krav, nya önskemål som kanske har uppkommit, samt alternativa lösningar. Dessa lösningar måste vara inom de ramar som utvecklaren har fått gällande begränsningar av systemet, speciellt de gällande resurser. Ledningen måste fastställa affärsfördelarna man kan få av det nya systemet. Systemutvecklarna skapar en grov uppskattning av kostnaden för att implementera systemet. Slutligen tas ett beslut på dessa premisser, om det är någon mening med att fortsätta.

### Systems Investigation / Systemutredning

Om ledningen är positiv till att fortsätta, blir nästa steg att göra en detaljerad rapport genom att undersöka:

- De funktionella krav man har på det existerande systemet, och utifall dessa krav uppfylls.
- De krav man har på det nya systemet eftersom det kan uppkomma nya situationer eller möjligheter vilka kanske uppmanar till ändrade krav.
- Begränsningar
- Den omfattning och volym av data som måste processas.
- Undantagshantering
- Problem med den nuvarande arbetsmetoden

Denna information blir mycket mer detaljerad än den information som rapporterades i förstudiefasen. Informationen fås genom intervjuer med personal, både på lednings och operationell nivå.

### Systems Analysis / Systemanalys

Med all fakta som insamlats fortsätter systemanalytikern att analysera det aktuella systemet genom att ställa följande frågor:

- Varför existerar problemen?
- Varför användes vissa arbetsmetoder?
- Finns det alternativa metoder?

Detta är ett försök att förstå alla aspekter av det aktuella systemet och varför systemet utvecklades som det gjorde, och senare kanske indikera hur saker kan förbättras genom ett nytt system.

### Systems Design / Systemdesign

Denna fas involverar både skapandet av själva systemet, samt en dokumentation. Om en grundlig undersökning gjord och de rätta frågorna har ställts i de tidigare faserna, skapas ett system som överensstämmer med designen man tänkt sig.

### Implementation / Implementation

I detta steg är det viktigt att alla aspekter av det nya systemet är testade. En viktig del i denna fas är kvalitets kontroll, den manuella proceduren liksom hård och mjukvara måste testas tills både analytikern och användaren är nöjd. Ytterligare en viktig aspekt i denna fas är utbildning och träning av användarna. Accepteringstestet spelar en nyckelroll i detta skede, om det inte passeras, så fördröjs hela projektet tills det passeras.

### Review and Maintenance / Granskning och drift

Det sista steget i systemutvecklingsprocessen sker när systemet är i drift. Det sker även en granskning av systemet för att försäkra sig att det uppfyller de krav som sattes i första fasen, samt att kostnaden inte överstiger den uppskattade.

#### *3.2.2 Potentiella svagheter med the traditional life cycle:*

Det har vuxit fram kritik till detta sätt att utveckla system, eller snarare sättet att använda metoden under årens lopp, och kritiken riktar sig framför allt på metodens svagheter inom dessa områden:

#### *Misslyckat att möta ledningens behov*

Systemen används nästan uteslutande till att lösa rutin och repeterande uppgifter. Istället för att möta företagets och ledningens behov, så används systemen till att hjälpa till att lösa operationella uppgifter.

#### *Entydig systemdesign*

Datasystem ersätter ofta manuella system, vilket har visats olämpligt när det gäller ändrade förhållanden. Systemanalytikerna pratar om att datorisera de manuella systemen och det är därför inte överraskande att den nya designen är entydig. Mer radikala och möjligen mer fördelsaktiga datasystem implementeras inte.

#### *Ostabila modeller av processer*

Allt eftersom datasystemmodeller processas, måste de modifieras eller frekvent skrivas om. Det kan därför sägas att datasystem, vilka är modeller av processer, är ostabila därför att verklighetens processer är ostabila.

#### *Outputdriven design leder till oflexibilitet*

Outputen som systemet är tänkt att producera bestäms tidigt i processen, därför kan ändrade krav av outputen kräva förändringar i designen, och detta i sin tur leder till förseningar av implementeringen, eller att man implementerar ett system som inte uppfyller kraven, eller helt enkelt är olämpligt.

#### *Missnöjda användare*

Ibland förkastas system så fort som de har implementerats, det kanske är första gången användaren ser resultatet av sina "beslut" som togs tidigt i processen när informationen var knapphändig. Det kan finnas ett gap mellan utvecklarna och användarna, vad gäller kunskap om teknologin och dess potential, och därför kan användarna inte aktivt delta i designen av

systemet. Detta gap kan i vissa fall leda till att användarna utvecklar ett eget informellt system som gör det tänkta systemet överflödigt.

#### *Problem med dokumentation*

Dokumentationen är i många fall inriktad till utvecklarna och inte till användarna, programmerarna ser det som ett krav av dess avdelning att göra en dokumentation. I värsta fall sker ingen modifiering av dokumentationen, då det skett sena ändringar i designen, eller i underhållsfasen.

#### *Brist på kontroll*

Trots att tillvägagångssättet med metoden gör det möjligt att uppskatta tid, människor och andra resurser, så blir dessa uppskattningar otillförlitliga på grund av komplexiteten i vissa av faserna, samt den osäkerheten av de som gjort uppskattningarna.

#### *Ofullständiga system*

Undantagshandling ignoreras, eftersom det tar för mycket tid och pengar. Dessa undantag indikerar både på ett tekniskt problem, men även ett systemfel nämligen det att inte analysera och designa systemet grundligt.

#### *Eftersläppande applikationsarbete*

Det kan vara flera system som väntar på att bli utvecklade, det kan i värsta fall ta ett år eller mer innan systemet börjar utvecklas, detta kan leda till att man skjuter upp krav eftersom man vet att det kommer att ta så pass lång tid.

#### *Upprätthålla arbetsbördan*

Eftersom man snabbt vill ha systemet i gång, kan det leda till att man gör "quick and dirty" lösningar, och detta har i sin tur lett till problemet med att många operationella system inte blir långlivade. I många fall tilldelar man så mycket som 60-80 procent av arbetet till att underhålla systemet.

#### *Problem med det ideala tillvägagångssättet*

Den traditionella SDLC modellen förutsätter en sekventiell och top-down process, detta är lite naivt. Det är nödvändigt att utföra iterationer t ex.. när nya krav dyker upp eller när man upptäcker att vissa subfaser är onödiga.

Grundorsaken till mjukvaruutvecklingens problem:

- Ad hoc kravhantering
- Oklar och ofullständig kommunikation
- Spröd arkitektur
- Överväldigande komplexitet
- Oupptäckt inkonsekvens i krav, design, och implementation
- Otillräcklig testning
- Subjektiv utvärdering av projektets status
- Misslyckande att attackera risker
- Okontrollerad spridning av förändringar
- Otillräcklig automatisering



### 3.3 Spiralmodellen

Spiralmodellen skapades av Boehm 1988, och istället för att representera mjukvaruutvecklingen i sekvenser av aktiviteter, så representeras processen som en spiral. Varje loop i spiralen representerar en fas i utvecklingen. Varje loop i spiralen är indelad i fyra sektioner och dessa är:

#### 1. Objective setting / Målformulering

Specifika mål för denna fas av projektet definieras. Begränsningar vad gäller både produkt och process identifieras, och en detaljerad plan görs. Risker identifieras, alternativa strategier, beroende på dessa risker planeras.

#### 2. Risk assesment and reduction / Riskbedömning

För varje identifierad risk i projektet görs en detaljerad analys. Steg tas för att minska risken, exempelvis om det finns en risk vad gällande krav, kan ett prototypsystem skapas.

#### 3. Development and validation / Utveckling och validering

Efter risk utvärdering väljs en utvecklingsmodell för systemet. Exempelvis om den största risken gäller subsystem integration så väljs vattenfallsmodellen dvs. det traditionella sättet som utvecklingsmodell.

#### 4. Planning / Planering

Projektet utvärderas och ett beslut tas om man ska fortsätta med ytterligare en loop i spiralen. Om ett beslut tas om fortsättning, görs en plan för nästa fas i projektet.

En viktig skillnad mellan spiralmodellen och andra mjukvaruprocess modeller är beaktningen av risker i spiralmodellen. Det finns inga statiska faser som specifikation eller design i spiralmodellen. Spiralmodellen använder sig av andra processmodeller. Prototyping kan användas i en spiral för att lösa osäkerhet rörande krav och för att minska risker. Detta kan sedan följas av en konventionell vattenfallsutveckling. (Sommerville 2001)

## 4 VAD ÄR RUP?

*I detta kapitel beskrivs vad RUP är och dess uppbyggnad.*

Rational unified process är en mjukvaruutvecklingsprocess skapad och marknadsförd av Rational Software. RUP har ambitionen att tillhandahålla ett disciplinerat tillvägagångssätt för att fördela arbetsuppgifter och ansvarsområden inom en utvecklings organisation. Målet med processen är att inom rimlig tid och budget, skapa högkvalitativ mjukvara som möter slutanvändarnas behov. Rational unified process innefattar många ”best practices” inom modern mjukvaruutveckling och presenterar dem i anpassad form som passar många projekt och organisationer. Följande definition av RUP baseras på Kruchten (Kruchten 2000) samt en utbildningspärm (RUP - en introduktion 2000).

### Bakgrund RUP

Objectory process skapades av Ivar Johansson 1987 och fokuserade på objekt orienterad design och användningsfall. Rational softwarecorporation och objectory AB slog sig samman och bildade Rational objectory process 1995. Rational unified process är efterföljaren som tillkom 1998.

### 4.1 Best practices

Med begreppet best practices avses kommersiellt beprövade tillvägagångssätt inom mjukvaruutveckling som, när de används kombinerat, angriper de grundläggande problemen inom mjukvaruutvecklingen. De är ”best practices” inte så mycket för att du precis kan kvantifiera deras värde, utan istället därför att de vanligtvis används av framgångsrika organisationer. Dessa ”best practices” är följande:

#### 4.1.1 Arbeta iterativt

Försöker identifiera risker tidigt i processen, när det är möjligt att ta hand om dessa risker effektivt och i tid. Detta tillvägagångssätt gör det möjligt att ha kontinuerlig utforskning, invention, och implementation, varje iteration tvingar utvecklingsteamet att driva fram projektets färdiga artefakter på förutsatt och repeterbart sätt. Genom att arbeta iterativt genereras en mängd lösningar på de problem man tidigare haft inom mjukvaruutvecklingen.

- Allvarliga missförstånd görs ofta tidigt i livscykeln, då det fortfarande är möjligt att ta hand om dessa.
- Detta tillvägagångssätt gör det möjligt och uppmanar till användare feedback för att få reda på systemets riktiga krav.
- Kontinuerlig, iterativ testning frambringar en objektiv bedömning av projektets status.
- Teamet kan lära sig av sina misstag och därför kontinuerligt förbättra processen.
- Intressenter kan få konkreta bevis på projektets status genom hela livscykeln.

Utvecklingsteamet tvingas fokusera på de kritiska frågorna för projektet och skyddas ifrån de som kan distrahera dem ifrån de riktiga riskerna.

Inkonsekvens i krav, design, och implementation upptäcks tidigt.

#### 4.1.2 Hantera kraven

Är ett systematiskt tillvägagångssätt för att framkalla, organisera, kommunicera, och ta hand om de förändrade krav som uppkommer. Att hantera kraven innebär att man ska dokumentera systemets krav på funktionalitet och begränsning, utvärdera förändringar för dessa krav, uppskatta deras påverkan, och hitta och dokumentera kompromisser och beslut. För att kunna hantera kraven på projektet finns det lösningar:

- Disciplinerat tillvägagångssätt byggs in i kravhanteringen.
- Kommunikation är baserad på definierade krav.
- Krav kan bli prioriterade, filtrerade, och upptäckta.
- Inkonsekvens upptäcks lättare.

Det finns fördelar med att effektivt hantera kraven och detta leder till att man får bättre kontroll på komplexa projekt. Ökad kvalitet och kundnöje uppnås när alla intressenter har samma grunduppfattning av vad som ska byggas och testas. Att upptäcka och hantera fel i kraven tidigt leder till lägre kostnad för projektet, samt att eventuell försening kan undvikas. Kravhantering underlättar inblandning av användarna att tidigt i processen, så att systemet uppfyller deras krav. En bra kravhantering bygger på en gemensam syn på projektets krav, och engagemang av alla inblandade parter som intressenter, kunder, ledning, designers samt testarna.

#### 4.1.3 Utveckla komponentbaserat

Visualisera, specificera, konstruera, och dokumentera en mjukvara kräver att systemet kan ses ur olika perspektiv. Alla intressenter har olika syn på projektet under dess livstid. Ett systems arkitektur är kanske det viktigaste att visa upp för att hantera dessa olika synsätt, för att kunna kontrollera den iterativa och inkrementella utvecklingen av ett system genom dess livscykel. Arkitekturen innefattar inte bara struktur och uppträde utan även användarvänlighet, funktionalitet, prestanda, återanvändning mm. Komponentbaserad utveckling är en viktig del inom mjukvaroarkitekturen eftersom den gör det möjligt att återanvända komponenter från tusentals tillgängliga källor. Ihopkopplat med att systemet utvecklas iterativt använda komponentbaserat arkitektur involverar kontinuerlig utveckling av ett systems arkitektur. Varje iteration producerar en exekverbar arkitektur som kan mätas, testas, och utvärderas mot systemets krav. Att använda komponentbaserad arkitektur ger lösningar på flera problem:

- Komponenter underlättar elastiska arkitekturer.
- Att använda moduler frambringar en klar separation av bekymmer mellan elementen i det system som ska ändras.
- Återanvändning underlättas genom standardiserade ramverk och kommersiellt tillgängliga komponenter.
- Visuella modelleringsverktyg tillhandahåller automatisering för komponentbaserad utveckling

#### 4.1.4 Modellera visuellt

Modellering är viktigt därför att det hjälper utvecklingsteamet att visualisera, specificera, konstruera, och dokumentera strukturen och beteendet av ett systems arkitektur. Genom att använda sig utav ett standard modellerings språk UML (Unified Modeling Language), kan medlemmar i utvecklingsteamet otvetydigt kommunicera med varandra. Visuella modelleringsverktyg underlättar kontrollen av dessa modeller, de låter dig gömma eller visa upp detaljer efter behov. Visuell modellering hjälper även till att upprätthålla kontinuerlighet mellan systemets artefakter, dess krav, design, och implementation. Kort sagt, visuell modellering hjälper till att öka teamets möjlighet att hand om systemets komplexitet. Genom att modellera visuellt ges olika lösningar på problem:

- Usecases och scenarios gör att man kan göra otvetydiga specificeringar.
- Modellers otvetydighet fångar mjukvarudesign
- Icke modulära och inflexibla arkitekturer exponeras
- Detaljer kan gömmas
- Applikationskvalité börjar med bra design
- Visuellt modelleringsverktyg ger stöd för UML modellering.

#### *4.1.5 Verifiera kvalitén kontinuerligt*

Det är väldigt viktigt att verifiera kvalitén i förhållande till dess funktionalitet, reliabilitet, applikation och systemprestanda. Verifiera ett systems funktionalitet involverar att skapa tester för varje enskilt nyckelscenario, där varje representerar någon aspekt av systemets önskade beteende. Man kan komma åt ett systems funktionalitet genom att fråga vilka scenarios som misslyckades och var. När du skapar din mjukvara iterativt, så testas du vid varje iteration, en process av kontinuerlig, kvantitativ utvärdering. Genom att verifiera mjukvarans kvalitét ges dessa lösningsförslag på problemen:

- Ett projekts status utvärderas objektivt, eftersom det är testresultat, och inte pappersdokument som utvärderas.
- Denna objektiva utvärdering exponerar inkonsekvenser i krav, design, och implementation.
- Testning och verifiering fokuseras på områden med hög risk, därför ökar kvalitén och effektiviteten på dessa områden.
- Defekter upptäcks tidigt vilket minskar kostnaden att ta hand om dessa.
- Automatiserade testverktyg tillhandahåller testning för funktionalitet, reliabilitet och prestanda.

#### *4.1.6 Hantera förändringar*

Koordinera aktiviteterna och artefakterna hos utvecklarna och teamen, involverar skapandet av upprepade arbetsflöden för att kunna kontrollera förändringar i mjukvaran och andra skapade artefakter. Denna koordination gör att det blir en bättre allokering av resurser baserat på projektets prioritet och risk, och verksamt tar hand om dessa förändringar genom iterationerna. Ihopkopplat med att skapandet sker iterativt, ger detta sätt dig möjligheten att kontinuerligt kontrollera förändringar, så att du verksamt kan hitta dem och sedan reagera på problemen. Genom att kontrollera och hantera förändringar ges dessa lösningar till problemen:

- Arbetsflödet av kravförändringar definieras och upprepas.
- Förändrade krav underlättar klar kommunikation.
- Isolerade arbetsplatser reducerar inblandning mellan teammedlemmar som arbetar parallellt.
- Förändringsgradsstatistik tillhandahåller en bra måttstock (metrics) för objektiv utvärdering av projektets status.
- Förändrings-spridning är tillgänglig och kontrollerad.

## 4.2 RUP:s uppbyggnad

Processtrukturen: Två dimensioner

Den horisontella axeln representerar tid och visar livscykelns aspekter på utvecklingsmetoden alltmedan den utvecklas. Den horisontella dimensionen motsvarar de dynamiska aspekterna av processen, den uttrycks i termer av cykler, faser, iterationer och milstolpar.

Den vertikala axeln representerar kärnan i arbetsflödet, som grupperar aktiviteter logiskt. Den vertikala dimensionen motsvarar de statiska aspekterna av processen, den uttrycks i termer av processkomponenter, aktiviteter, arbetsflöden, artefakter och roller.

### 4.2.1 Den horisontella dimensionen

RUP bygger på att använda sig av den iterativa processen. När man använder sig av RUP vid framtagandet av ett system går man igenom RUP: s fyra faser. De fyra faserna är Inception, Elaboration, Construction och Transition och beskrivs nedan.

I varje fas skall vissa artefakter påbörjas, bearbetas eller avslutas. Dessa artefakter blir milstolpar för vad utvecklaren måste ha åstadkommit i fasen innan han går vidare till nästa fas. Milstenarna blir delmål som är till för att hjälpa utvecklaren att bedöma om projektet är moget för att gå vidare med det till nästa fas i utvecklingen. Grundtanken med uppdelningen i faser är att de skall hjälpa till att upprätthålla hög kvalitet då de skall underlätta krav och riskhantering tidigt i projektet.

De fyra faserna:

*Inception-fasen*



Målen med Inceptions-fasen är:

- Att bli överens om vad projektet ska åstadkomma, och dess begränsningar.
- Planer på en hög nivå skapas för hela projektet.
- Bestämma kritiska användarfall och primära scenarios som kommer ge de största design fördelarna.
- Demonstrera en kandidatarkitektur mot någon av de primära scenarierna.
- Uppskatta den totala kostnaden och schemalägga hela projektet
- Uppskatta risker.

Aktiviteter

Formulera projektets omfattning, fånga kontexten och de viktigaste kraven och begränsningarna för att få fram kriterier för slut produkten.

Planera och förbereda ett affärsfall och utvärdera alternativ för risk, personal, projektplan.

Skapa en kandidatarkitektur.

Utkomst

Visionsdokument, generell version gällande kärnan av krav i projektet, nyckeldrag, samt de största begränsningarna.

Användningsfallsmodell, undersökning med alla användningsfall och aktörer.

Initial projektordlista

Initial affärsfall, vilken innehåller affärs kontext, framgångskriterier, finansiell uppskattning.

Initial risktilldelning

Projekplan som visar faserna och iterationerna.

Milestone: lifecycle objectives(LCO)

### *Elaboration-fasen*



Målen med Elaborations-fasen är:

- Definiera, validera grundversionsarkitekturen så snabbt som är praktiskt möjligt.
- Skapa en grundversion
- Grundversion av en detaljerad plan för konstruktionsfasen
- Demonstrera att grundversion arkitekturen kommer att stödja visionen till en accepterad kostnad och tid.

Aktiviteter

Skapa visionen och de mest kritiska användarfallen som driver fram arkitektur och planerings beslut.

Utreda processen och infrastrukturen av utvecklingsmiljön

Utreda arkitekturen och välja komponenter.

Utkomst

En användningsfallmodell (80% färdig)

Kompletterande krav

En exekverbar arkitekturprototyp

Granskade affärsfall och en risklista

Utvecklings plan för hela projektet

En preliminär användarmanual

Milestone: Lifecycle Architecture (LCA)

### *Construction-fasen*



Målen med Constructions-fasen är:

- Minimera utvecklingskostnad genom att optimera resurser och undvika onödiga fragment och omarbete.
- Uppnå adekvat kvalitet så snabbt som är praktiskt.
- Uppnå användbara versioner (alfa, beta, och andra test releaser) så snabbt som möjligt.

Aktiviteter

Ledning och kontroll av resurser, och processoptimering

Göra färdigt komponentutveckling och testa mot de definierade utvärderade kriterierna.

(Avvägning av produktsläpp mot acceptanskriterier i visionen).

Utkomst

Mjukvaruprodukten, integrerad på adekvat plattform

Nödvändig användarmanual

En beskrivning av den aktuella releasen.

Milestone. Initial operational capability(IOC) ”betarealise”

*Transitions-fasen*



Målen med Transitions-fasen är:

- Uppnå möjlighet till att användaren själv kan få support.
- Uppnå intressenters samstämmighet att driftsättning av grundversionen är komplett och konsistent med de utvärderade kriterierna i visionen
- Uppnå slutgiltig produktgrundversion så snabbt och kostnads effektivt som möjligt.

Aktiviteter

Driftsättnings - specifika arbetsuppgifter så som kommersiell packning och produktion.

Avstämningsaktiviteter som att fixa buggar.

Jämföra den driftsatta grundversionen mot visionen och acceptanskriterier för produkten.

Utkomst

Det färdiga systemet

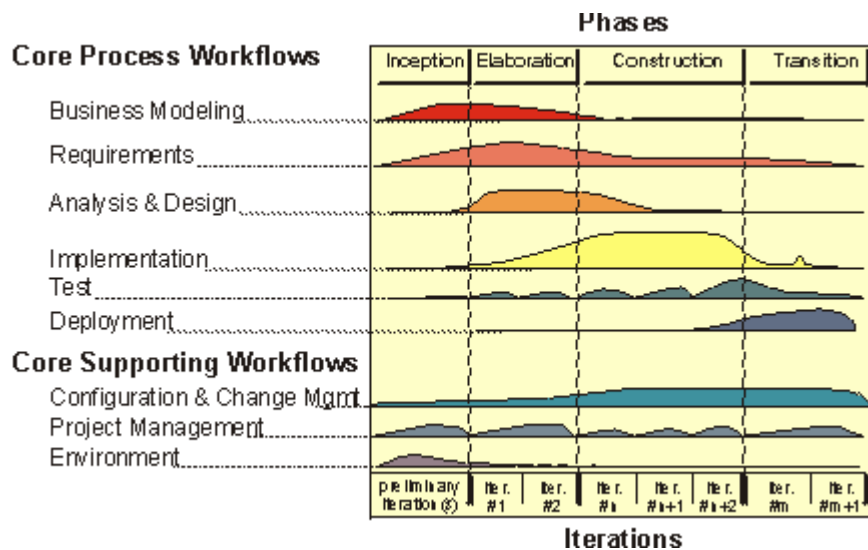
Milestone: product release(GA)

#### 4.2.2 Den vertikala dimensionen

##### Huvudflöden

Arbetsflöde - när ska det utföras?

Ett arbetsflöde är en sekvens av aktiviteter som producerar ett resultat av observerbart värde. Det finns många sätt att organisera de uppsättningar aktiviteter i ett arbetsflöde, i RUP använder man sig av Kärnprocessarbetsflöden – det finns sex olika kärnprocessarbetsflöden i RUP, och tre stödjande arbetsflöden. De representerar en uppdelning av alla roller och aktiviteter i logiska grupperingar. Dessa arbetsflöden beskrivs nedan.



Figur 2 processöversikten (Kruchten, 2000)

##### Business modeling / Affärs modellering

Syftet med flödet är att utvecklare och intressenter skall förstå strukturen och dynamiken hos den organisation i vilken systemet skall implementeras. Man bör få en förståelse för de nuvarande problemen i organisationen, och identifiera möjliga förbättringar. Ta fram de systemkrav som behövs för att stödja organisationen. Se till att utvecklarna och kunden har en gemensam uppfattning av organisationen

##### Requirements / Krav

Här är målet att alla intressenter skall ha en överenskommelse över vad systemet skall kunna utföra. Man definierar systemets gränser. Det skapas en uppskattning av kostnad och tidsåtgång för utvecklingen av systemet. Det tas fram underlag för planering av iterationernas tekniska innehåll. Ett användargränssnitt definieras för systemet som fokuserar på användarnas behov.

##### Analysis & Design / Analys & Design

Syftet är att omvandla krav på systemet till design av det blivande systemet. Utvecklaren bör få fram hur kraven skall implementeras, samt anpassa designen till implementationsmiljön. Man utvecklar här en robust arkitektur för systemet, så att du kan designa ett system som är enkelt att förstå, bygga, och utvecklas. Den primära artefakten av analys och designflödet är



designmodellen. Denna modell består av en uppsättning samarbetande modellelement som visar på systemets beteende.

#### Implementation / Implementation

Här definieras hur källkoden i systemet skall organiseras t ex. så väljer man i vilka lager olika subsystem skall ligga. Komponenter byggs och implementeras som klasser och objekt. Komponenterna testas sedan som enheter och sedan sätter man samman de olika delarna till ett exekverbart system.

#### Test / Test

Kontroll sker så att alla krav har blivit korrekt implementerade, och att alla objekt agerar som de skall. Man kontrollerar även att alla komponenter och objekt har integrerats på ett korrekt sätt i systemet. Identifiera och klargör att alla upptäckta defekter har tagits hand om innan mjukvaran sätts i drift.

#### Deployment / Driftsättning

Här beskrivs de aktiviteter som skall säkerställa att slutprodukten blir tillgänglig för dess slutanvändare. Testar mjukvaran i sin operationella miljö ett sk. beta-test. Packning av mjukvaran och leverans sker även här. När man har installerat systemet så tränar man slutanvändarna på det nya systemet.

De stödjande arbetsflödena är följande:

#### Configuration & Change management / Konfiguration & förändringshantering

Syftet med detta arbetsflöde är att bibehålla integriteten av projektets artefakter allteftersom de utvecklas i respons till ändrade önsningar. Konfigurationshantering innefattar produktstruktur, identifikation av element, och versioner. Förändringshantering av önskemål omfattar processen av modifieringar av artefakter på ett konsekvent sätt. Arbetsflödet löper genom hela utvecklingsprocessen, och det syftar till att kontrollera versioner av dokument och kod, samt förändringar i kravbilden.

#### Projekt management / Projektledning

Tillhandahåller ett ramverk för hantering av intensiva mjukvaruprojekt. Här utvärderas möjligheter och risker i projektet. Man sköter utvecklingsplanering, övervakning och kontroll av projektet, samtidigt som man ser till att den iterativa processen följs och att projektet slutförs. Tillhandahålla ett ramverk för riskhantering.

#### Environment / Miljö

Detta arbetsflöde hanterar hur man kan anpassa processen till organisationen. Man vill förse utvecklingsorganisationen med en utvecklingsmiljö dvs. både process och verktyg, vilka skall stödja utvecklingsteamet i hela processen. Teknisk tjänster för att stödja processen är informationsteknologiinfrastruktur, kontoadministration, back up osv..

Varje kärnarbetsflöde täcker en stor grund, för att bryta ner dem använder sig RUP av arbetsflödesdetaljer för att upptäcka en specifik grupp av nära relaterade aktiviteter. Aktiviteterna utförs tillsammans eller i en cyklisk form, eller utförs av en grupp av människor som arbetar tillsammans i en workshop. Arbetsflödesdetaljer visar även informationsflöde – artefakterna som är input till och output från aktiviteterna – visar hur aktiviteter intertakterar genom de olika artefakterna.

Iterationsplaner är ett annat sätt att presentera processen, beskriva den med ett perspektiv utifrån vad som händer i en typisk iteration. Man kan titta på dem som en ögonblicklig bild av processen för en given iteration, man kan välja den aktivitet som kommer att köras under iterationen, och kopiera dem när det är nödvändigt.

#### 4.2.3 Processens statistiska struktur:

Workers / Roll - vem utför uppgiften?

Detta är det centrala konceptet i processen, en roll definierar arbete och det ansvar en individ eller en grupp av individer har som arbetar tillsammans som ett team. Arbetet uttrycks i termer av aktiviteter som rollen utför, och varje roll är associerad med en uppsättning sammanhängande aktiviteter, och det är de aktiviteter som bäst utförs av denna person. Det hjälper att tänka sig en roll som en hatt som individer bär under projektets gång. En person kan bära flera hattar. Det är viktigt att göra en distinktion mellan roll och en person, termen roll refererar till de roller som definierar hur individer ska utföra sitt arbete. En roll utför en eller flera roller och är ägare till en uppsättning av artefakter.

Exempel på roller:

Systemanalytiker, designer, testdesigner.

Projektledaren fördelar rollerna när han planerar projektet, denna fördelning gör det möjligt att en individ kan arbeta som flera roller och för en roll att spelas av flera individer.

Aktivitet - hur ska uppgiften utföras?

Roller har aktiviteter, vilka definierar det arbete de utför. En aktivitet är en samling av arbete som en individ i dess roll kan ombedjas att utföra, och det produceras ett meningsfullt resultat i kontext med projektet. Aktiviteten har ett klart syfte, ofta uttryckt i termer av skapa eller uppdatera artefakter, som en modell, en klass, eller en plan. Varje aktivitet är tilldelad till en specifik roll. Aktiviteter kan upprepas flera gånger på samma artefakt, speciellt genom en iteration till en annan allteftersom systemet förbättras och utökas. Upprepade aktiviteter kan utföras av samma roll, men inte nödvändigtvis av samma individ.

Exempel på aktiviteter:

Planera en iteration, hitta use cases och aktörer, exekvera en test om systemets prestanda.

Aktiviteter delas in i steg, dessa steg kategoriseras som följer:

- Tänkande – rollen försöker förstå uppgiftens natur, samlar och undersöker input artefakterna, och formulerar outputen.
- Utförande – rollen skapar eller uppdaterar några artefakter.
- Undersökande – rollen inspekterar resultatet mot uppsatta kriterier.

Artefakt - vad ska utföras?

Aktiviteter har input och dess output blir artefakter. En artefakt är en mängd information som produceras, modifieras, eller används av en process. Artefakter är den påtagliga produkten i projektet, de saker som projektet producerar eller använder under tiden man arbetar mot den slutgiltiga produkten. Artefakter används som input till arbetare för att utföra en aktivitet, och är resultatet eller output av denna aktivitet.

Exempel på artefakter:

- Modell – use-casemodell eller design modell
- Ett modellelement- ett element inom en modell, en klass, en use-case, eller ett subsystem.
- Ett dokument- affärsfall eller mjukvaruarkitektur dokument.

- Källkod

Artefakterna i RUP har organiserats i fem olika informationsdelar:

Management set	Grupperar alla artefakter relaterade till software business, och till ledningen av projektet. Planerade artefakter, de som software development plan (SDP), business case. Osv..
Requirement set	Grupperar alla artefakter som rör definitionen av den mjukvara som ska skapas. Visionsdokumentet, krav vad gällande intressenter, use-case modell.
Design set	Innehåller en beskrivning av det system som ska byggas. Designmodell, arkitekturbeskrivning, testmodell.
Deployment set	Käll och exekveringskod, associerade datafiler.
Implementation set	Innehåller all den information som levereras inkluderat, installationsmaterial, användarmanual, träningsmaterial.

#### Tilläggande processelement

Roller, aktiviteter, och artefakter representerar ryggraden i RUPs statiska struktur. Men andra element läggs till aktiviteter eller artefakter för att göra processen lättare att förstå och använda, och för att tillhandahålla mer guidning för användaren.

Dessa är:

- *Riktlinjer* – är regler, rekommendationer, eller heuristik som stödjer aktiviteter och steg. De beskriver välformade artefakter, fokuserar på dess specifika kvalitéer.
- *Mallar* – är modeller eller prototyper av artefakter. Mallarna är anpassade till det verktyg som ska användas ex. Microsoft Word mallar för dokument eller rapporter med företagets logga.
- *Verktyg mentorer* – visar hur du utför stegen genom att använda ett speciellt verktyg ex. Rational Rose.
- *Koncept* – Några av nyckelkoncepten, som iterationer, faser mm. Introduceras i separata delar av processen, vanligtvis sammankopplat med lämpligt arbetsflöde.

## 5 PROCESSFÖRBÄTTRING

*I detta avsnitt beskrivs utifrån vad tre olika forskare Sommerville, Hägerfors, Checkland & Scholes anser karaktärisera en process, samt vad en framgångsrik process ska innehålla.*

Jag har valt dessa tre eftersom de alla tre har olika perspektiv på processen. Först ut är Sommerville och enligt han karaktäriseras en process av följande:

### 5.1 Sommerville

- **Understandability – förståelighet**  
En process ska vara explicit definierad och det ska vara enkelt att förstå process definitionen.
- **Visualibility – synlighet**  
Processaktiviteterna ska ge klara resultat så att framåtskridandet av processen externt blir synlig.
- **Supportability – stödjande**  
Processaktiviteterna ska kunna stödjas med tekniker och verktyg.
- **Acceptability – grad av acceptans**  
Den definierade processen ska vara accepterad av, och användbar för utvecklarna.
- **Reliability – reliabilitet**  
Processen ska vara designad så att processfel undviks eller blir funna innan de resulterar i produktfel.
- **Robustness – robusthet**  
Processen ska kunna fortsätta trots oväntade problem.
- **Maintanability – varaktighet**  
Processen ska kunna utvecklas för att reflektera förändrade organisationskrav eller identifierade processförbättringar.
- **Rapidity – snabbhet**  
Processen ska snabbt kunna leverera ett system från en given specifikation.

Sommerville menar att det är inte möjligt att förbättra processen som optimerar alla processattribut samtidigt, exempelvis om processen har hög grad av synlighet kräver detta i många fall mycket dokumentation och detta gör processen långsammare. (Sommerville 2001)

### 5.2 Hägerfors

Hägerfors ställer upp ett antal påståenden som hon anser en process av hög kvalitet ska innehålla

- Att välja tekniker och metoder för att utföra arbetet så att de överensstämmer med situationen, organisationen, personer och mål.
- Att teknikerna och metoderna används på ett kompetent sätt.
- Att fördela arbetet på ett bra sätt med hänsyn till tillgängliga personer.

- Att all behövlig kompetens finns tillgänglig.
- Att personer kan arbeta och trivas tillsammans. (Hägerfors 1995)

### 5.3 Checkland & Scholes

Petter Checkland och Jim Scholes som arbetar med Soft system methodology har utvecklat kriterier som de använder för att värdera processer. Kriterierna kallas de 5E:na på grund av deras gemensamma begynnelse bokstav. ( Checkland & Scholes 1990)

*Efficiency* – Fungerar medlen t ex. valda tekniker? Nås den förändring med dessa medel? Fungerar processen? Resulterar processen i den önskade produkten?

*Efficiency* – Används minimala resurser? Finns det andra sätt att åstadkomma förändringen, som är billigare t ex. andra tekniker?

*Effectiveness* – Leder medlen till uppnående av långsiktiga mål t ex. ökad marknadsandel eller god arbetsmiljö?

*Ethicality* – Är det moraliskt rätt att göra denna förändring? Uppfattas processen som bra eller dålig?

*Elegance* – Tilltalar denna förändring vårt skönhetsinne? Leder förändringen till att verksamheten fungerar bättre och smidigare? Är processen utformad på ett tilltalande sätt?

Dessa kriterier tar inte upp några sociala aspekter, inte heller hur kommunikation och information fungerar och kommer att fungera under processen.

## 5.4 Analysmodellen

Jag har i analysmodellen använt mig av tre olika forskares synpunkter och kriterier om vad som karaktäriserar en framgångsrik process. Ordningen är organiserad så att årtalen speglar utvecklingen av dessa kriterier, och senast kommer först.

	<b>Kriterier</b>
<b>Sommerville 2001</b>	<ol style="list-style-type: none"><li>1 <b>Understandability</b></li><li>2 <b>Visualibility</b></li><li>3 <b>Supportability</b></li><li>4 <b>Acceptability</b></li><li>5 <b>Maintanability</b></li><li>6 <b>Reliability</b></li><li>7 <b>Robustness</b></li><li>8 <b>Rapidity</b></li></ol>
<b>Hägerfors 1995</b>	<ol style="list-style-type: none"><li>1 <b>Att välja tekniker och metoder för att utföra arbetet så att de överensstämmer med situationen, organisationen, personer och mål.</b></li><li>2 <b>Att teknikerna och metoderna används på ett kompetent sätt.</b></li><li>3 <b>Att fördela arbetet på ett bra sätt med hänsyn till tillgängliga personer.</b></li><li>4 <b>Att all behövlig kompetens finns tillgänglig.</b></li></ol>
<b>Checkland 1990</b>	<ol style="list-style-type: none"><li>1 <b>Efficiacy</b></li><li>2 <b>Efficiency</b></li><li>3 <b>Effectivness</b></li><li>4 <b>Ethicality</b></li><li>5 <b>Elegance</b></li></ol>

## 6 EMPIRI

*I detta kapitel ges en kort beskrivning av respondenterna,*

Jag har valt att intervjua tre personer med olika erfarenhet av RUP. Detta för att se hur deras erfarenhet av att använda RUP påverkar deras syn på processen. Respondenterna har alla tre gedigen kunskap om RUP och detta krävdes för att kunna svara på mina frågor. Alla tre har adekvat utbildning och flerårig erfarenhet av systemutveckling och alla tre har tidigare arbetat med andra metoder.

- Petter har arbetat med systemutveckling i snart 15 år och har arbetat efter RUP sedan lanseringen och arbetar i dag som RUP mentor.
- Jens har flerårig erfarenhet av RUP och arbetande tidigare efter en liknande metod som även den var iterativ.
- Anna har använt metoden i flera olika projekt när hon tidigare arbetade som konsult.

Vad respondenterna svarade på respektive fråga kan läsas i appendix.

## 7 RESULTAT OCH ANALYS

*I detta kapitel analyseras resultatet ifrån intervjuerna och jämförs med teorin. Ordningen är strukturerad efter min analysmodell.*

### 7.1 Sommerville

#### 7.1.1 Understandability – förståelighet

För att kunna avgöra om processen är förståelig eller inte, ställdes frågor hur lätt det är att förstå processen, samt till vilken grad den är definierad. De intervjuade ansåg att RUP är väl definierad, och i vissa fall överdefinierad, och det i sådana fall krävdes kunskap för att kunna plocka ut det relevanta, för att använda sig av allt blir för stort. Petter ansåg att det fanns så många vägar in till detaljinformationen att detta kan ställa till problem att man till slut inte vet var man har hamnat. Jens anser att processen är enkel och självklar, att grundidéerna har använts i traditionella utvecklingsmetoder, så RUP är ingen revolution utan är en paketering av välkända faktorer på ett bra sätt, däremot att ordningen har adresserats om hur man angriper saker är den stora skillnaden.

Jag anser att svaren på dessa frågor ger svar på att processen är väl definierad och att det relativt enkelt att förstå processdefinitionen, men att det krävs kunskap och erfarenhet antingen från utbildning eller arbete i tidigare projekt för att kunna använda sig av processen på ett effektivt sätt, som Anna sa att det kräver erfarenhet för att kunna veta vad man ska plocka, för det är meningen att man ska anpassa processen till sitt projekt.

Fler frågor som jag försökte få svar på var hur förståelsen av processbegreppen är i RUP? Är det många återkommande begrepp ifrån tidigare metoder, har i så fall dessa begrepp gjort så att det blivit förbättringar/förenklningar? Petter anser att begreppen är väldigt tydliga, och tack vare upphovsmännen Jacobsson, Rumbaugh och Booch har slagit sina begreppsvärldar samman har det satt betydelsen.

Jens anser att förståelsen av processbegreppen är hög i de fall där personen är insatt i engelsk och amerikansk litteratur i ämnet, samt om man driver projekt där koncernspråket är engelska, problem kan det bli när ett företag har en svensk profil och vill använda sig av begreppen på svenska, då kan det bli problem med att skilja på begreppen person och roll, att samma person kan ha flera roller. Det är mottagaren av ett projekt som har svårast med detta.

Att RUP bygger på UML, där har RUP ett försprång gentemot många processer att det är en standard notation som man redan kan om man sysslar med utveckling i branschen, så den är lätt att förstå anser Anna.

Jag anser att förståelsen av processbegreppen i RUP är hög, och i vissa fall anses det som de facto standard, detta tror jag nog kommer att ta ett tag till innan det blir. Men allokeringen av begrepp ansågs positivt ifrån de jag intervjuade, trots att begreppen ansågs finnas redan, men att de ej tidigare varit samlade.

Jag tror nog att begreppsförståelsen har UML att tacka för mycket, för har man tidigare arbetat med denna notation känns nog de flesta begreppen igen. Att begreppen är samlade och många till antalet gör att komplexiteten ökar, och detta kräver återigen kunskap och erfarenhet, men kan man RUP och dess begrepp får man en mer fullständig begreppsvärld att plocka ifrån. När du väl förstår RUP är det mycket bra, och det är nog tack vare skaparnas långa erfarenhet och att de har hittat begrepp som saknades tidigare anser Petter.



En svårighet med begreppen kan vara att dels förstå vad de säger, sedan förstå vad de anser, dess intension. Detta kräver att man först förstår helheten för att kunna sätta sig in i detaljer. Att skraddasy begreppsvärlden för att göra det begripligt för exempelvis en kund kan vara att helt enkelt tar bort en massa information så att det bara finns en väg in till detaljerna.

Jag frågade vidare de intervjuade om de ansåg att kundförståelsen ökar med hjälp av att använda RUP?

Petter anser att kundförståelsen ökar eftersom RUP börjar bli de facto standard så att du kan börja relatera till samma metod. Tidigare har det ofta varit så att du har haft en metod och kunden en annan. De olika modellerna ex business arbetsflödet i RUP gör att det blir enklare att kommunicera med kunderna och Petter anser att gör du dina användningsfall rätt ger det väldigt mycket bättre kundförståelse.

Att använda sig av use-cases i kravflödet gör att man kan försöka göra en bild av systemets beteende, och få denna beskrivning så pass komplett och begriplig och visa denna för kund, att alltså från början visa på hur systemet kommer att fungera och upplevas. Återigen detta med bilder så använder sig RUP av UML diagram, och när organisationen lärt sig att kommunicera genom dessa får du mycket bättre förståelse anser Petter.

Anna anser att de olika arbetsflödena hjälper till att förstå vad kunden vill ha, samt att hjälpa kunden att veta vad den vill ha.

Jens anser att förståelsen från kund är sämre med RUP än någon traditionell metod. Om man sedan ser på kundens verksamhet, eller vad det är som ska lösas så tror Jens att man når en högre detaljnivå fortare inom riskområdena med hjälp av RUP.

Kundförståelsen anser jag är helt beroende på kundens mognad inom RUP, använder sig kundens organisation av processen ökar förståelsen mellan de inblandade parterna. De organisationer som inte använder sig av RUP anser nog att förståelsen försämras på grund av olika begrepp och definitioner. Men om kunden använder sig av RUP och dess modeller och notation ökar förståelsen mycket åt båda håll, både för att förstå vad det är kunden vill ha, samt vad den behöver. Jag tror att RUP kan vara användbar trots att kunden inte använder sig av metoden, speciellt i de fall kunden inte exakt vet vad det är han behöver.

Jag frågade respondenterna om kommunikationen underlättas mellan de inblandade i ett projekt, och i så fall hur?

Oklar och ofullständig kommunikation var ett av huvudproblemen med tidigare systemutvecklingsmetoder, RUP anser att detta problem kan undanröjas med hjälp av användandet av otvetydiga modeller, och genom användandet av dessa modeller underlätta kommunikationen mellan inblandade parter i systemutvecklingen. De jag har intervjuat har alla instämt att kommunikationen underlättas med hjälp av dessa modeller, och att notationen UML gör att alla pratar samma "språk". Arbetsflödet Analys och Design använder sig av roller i stället för en organisatorisk enhet, det kan ju vara samma person. Genom detta har uppfattningen underlättats om vad som behöver göras, och när, du har alltså frikopplat de olika arbetsuppgifterna från varandra.

Vidare så anser de intervjuade att det finns både en tydlighet i artefakterna och i vad man pratar om som gör det lättare att kommunicera. Tack vare att det finns en tydlig fokusering med modellernas så vet personen var han befinner sig i utvecklandet eller konstruktionen. Problem kan uppstå om det skapas tre läger anser en av de intervjuade, kravhantering,

programmering och databasmänniskorna måste alla tre tycka att det är intressant och effektivt att använda sig av samma modeller och notation för att kommunikationen ska fungera bra. Bättre kommunikation är nog en förutsättning för att RUP ska lyckas bli det Rational hoppas, för om inte detta sker känns begreppsvärlden och dess alla modeller ganska redundanta.

Slutligen för att avgöra om processen inom ett systemutvecklingsprojekt förbättras genom att använda sig av RUP angående dess förståelighet, och svara på om processen är definierad och förståelig, anser jag att utifrån vad teorin anser karakterisera en bra process och de kriterier jag ställt upp, samt de svar jag fick på mina intervjuer är att RUP är både bra definierad och förståelig. Dock att beakta är att utbildning och erfarenhet krävs för att förstå och kunna använda metoden på ett effektivt sätt, har man inte detta så kan metoden lätt kännas både stor och otymplig. Speciellt stor skillnad angående erfarenhet blir det vad gäller effektivitet i kommunikationen, som med erfarna användare av metoden underlättas mycket.

### *7.1.2 Visualibility – synlighet*

En av frågorna jag ställde för att få svar på om processens synlighet var om processens aktiviteter ger klara resultat så att framåtskridandet av processen externt blir synlig? De intervjuade anser att RUP är bra framme när det gäller artefakttänkandet, att varje aktivitet skapar en artefakt, och att metoden har sina milstolpar och uppdelningen i iterationer. Samt att man kommer överens med kunden vad som ska levereras vid första milstolpen. Detta gör att kunden kan se vad som sker inne i processen, att kunden får en artefakt levererad.

Nackdel kan vara att processen är så pass komplett att man leds till att tro att den är mer precis än vad den i själva verket är ex. aktiviteterna har tydliga resultat som ex vilket dokument som ska produceras men man måste själv tala om hur mycket som ska produceras av exempelvis en draft. Anna anser att det finns en risk med RUP att man bara dokumenterar och inte bygger, och att detta ansvar ligger mycket på projektledaren att se till att detta inte sker.

Jens svar går tvärtemot de andras och han säger att ett riskorienterat projekt med ganska mycket okända faktorer inte kommer att leverera speciellt mycket under den första tiden, och det gör ju att projektet snarare blir mer osynligt, i form av att det inte fysiskt visar vad som gjorts. Detta är en paradox jämfört med vad teorin säger om RUP, där ”Intressenter får konkreta bevis på projektets status genom hela livscykeln”. Denna skillnad i svar gentemot de andra intervjuade kan bero på att Jens arbetat främst som systemarkitekt och inte haft den direkta kontakten med kund, och således inte levererat artefakter efterhand.

För att kunna svar på utfall processen är synlig frågade jag om det finns en röd tråd som är synlig genom hela projektet?

Petter anser att den röda tråd som finns är dina arbetsflöden och den så kallade sega råttan bilden, där du talar om vad varje arbetsflöde ska producera och någon stans vid varje milstolpe så har du ju exekverings kriterier. Har jag uppfyllt dessa, blir detta den röda tråden när det gäller framåtskridande i processen. Petter har upplevt att vissa personer inom ett projekt har haft vissa svårigheter med att se vilken aktivitet som ska göras först, och vilka som kan göras parallellt. De har med andra ord haft svårigheter med att tolka RUP i det avseendet.

Jens tycker att RUP har en bra utgångspunkt i risklistan, det är den som är kritisk och styrande i hans arbete, med hjälp av denna vet han i vilken ordning han ska angripa saker. I de fall när kunden inte har samma insikt i sina system eller processen, anser Jens att RUP förutsätter nästan att processen är förankrad i kundens verksamhet för att anses som synlig.

Anna anser att processöversikten, kartan som man jobbar efter fungerar som en röd tråd genom projektet, hon planerar och följer upp projektet efter denna, det är en bild från projektet start till mål Även faserna och kärnarbetsflödena, samt milstolparna gör att hon anser att man kan följa utvecklingen. Anna säger att de även kommunicerar mot kunden med kartan och uppvisar status utifrån milstolparna. Detta är bara positivt och arbetar man i licensen kan man gå ner i de här olika arbetsflödena och hitta guidelines och mallar för de olika faserna, detta kräver dock ett bra metodstöd eller en väldigt kunnig projektledare, för att som utvecklare orkar du inte ta tag i det för att börjar man gräva sig ner så bara expanderar det.

Sammanfattningsvis upptäckte jag en skillnad i synlighet för den enskilde inom ett projekt, och synligheten gentemot kund, där åsikterna gick isär. En av de intervjuade ansåg att RUP måste till viss del vara förankrad i kundens organisation för att den ska vara synlig för kund, detta håller jag inte riktigt med om, det sker ju en överenskommelse om vad som ska produceras vid varje milstolpe, där tydliga kriterier sätts upp i samförstånd mellan utvecklare och kund och uppfylls dessa, så borde ju processen vara relativt synlig för kunden.

Ett problem i de tidigare utvecklingsmetoderna var bristen på kontroll pga. komplexiteten i faserna. Detta anser jag att RUP är bra på motverka med sina bilder och modeller, speciellt översiktskartan gör att man får bättre kontroll på var man befinner sig, samt vart man är på väg. Faserna är fortfarande komplexa men en av fördelarna med modeller är ju möjligheten att ”gömma” detaljer som inte anses vara kritiska för kunden att förstå.

Missnöjda användare var även det ett problem i tidigare metoder för det var i många fall första gången användaren ser resultatet av sina ”beslut” som togs tidigt i processen när informationen var knapphändig. Att det var ett gap mellan utvecklarna och användarna. Detta motverkar RUP på ett bra sätt genom sina iterationer och att varje aktivitet ger en artefakt.

### *7.1.3 Supportability – stödjande*

Ett av kriterierna var om processaktiviteterna kan stödjas med tekniker och verktyg?

Petter anser att det finns ett verktyg för allt i RUP, alla aktiviteter kan stödjas på ett eller annat sätt, det beror lite på hur mogen utvecklingsorganisation man är, det finns ett komplett utbud med tillräckligt bra verktyg. Man behöver inte hela konceptet med verktyg och tekniker utan man anpassar det efter projektets behov.

Detta stöd av tekniker och verktyg skiljer RUP ifrån tidigare metoder, där det inte fanns exempelvis verktyg så pass starkt anpassade och integrerade i processen.

De intervjuade är överens om att det finns i väldigt hög grad verktyg som stödjer processaktiviteterna, i vissa fall lite väl många, samt att arbetar man med Rationals leverans av RUP förutsätts det att man använder deras verktyg. Problem kan då uppstå utfall en kund skulle vilja byta ut en komponent i deras utbud, det kan exempelvis leda till kundberoende i det avseendet vilken versionshantering som används osv..

Vidare frågade jag om de anser du att processen blir bättre med hjälp av verktygen?

Petter anser att implementationen av processen blir lättare, verktygen underlättar arbetet med att samla ihop resultatet och det är ju alltid bra om du köper verktyg som är integrerade i processen.

Jens anser att det är ett krav på en bra utvecklingsprocess att man ska kunna välja verktyg självständig, att det inte är det en utvecklingsprocess säljer, alltså ett behov av ett verktyg, utan ett säkerhetsställande att jag utvecklar något på ett bra sätt, med kvalité i varje led där man jobbar.

Om det är hyfsad storlek på projektet blir processen blir bättre med hjälp av verktygen, men man måste var medveten om att för varje nytt verktyg man tar in så blir det en risk också anser Anna.

Sammanfattningsvis så anser jag att det finns verktyg som stödjer arbetet med processen, nackdelen är att det nästan förutsätter att man använder sig av Rational's verktyg, det är svårt att använda sig av andra verktyg eftersom dessa är så tydligt integrerade i processen, i viss utsträckning så lång att en av de intervjuade kallade RUP för verktygsleverantör.

En annan aspekt på användningen av verktyg i syfte att stödja processen är att det kostar väldigt mycket pengar att använda Rational's verktyg, så man måste noga analysera projektets behov av vilka verktyg som behövs. Verktygen anses kompletta men en av de intervjuade ansåg att det fattades verktyg för editering, komprimering och debuggning. Samt ett databashanteringssystem så man kan uttrycka det så att det finns tillräckligt med verktyg till processen , men inte till ett projekt.

Det finns även en inläringströskel för att effektivt använda dessa verktyg, de är väldigt sofistikerade. Verktygen är stödjande i det avseendet att det blir det lättare att arbeta i processen, och det finns en risk att om man inte skulle använda sig av verktygen, att man använder RUP som en projektlednings metod.

#### *7.1.4 Acceptability –grad av acceptans*

Är den definierade processen accepterad av, och användbar för utvecklarna? Processen accepterad av ledningen? Kunderna? Produktionsansvariga?

Petter säger att det beror på organisationen, att det är en mognadsfråga, i tekniska projekt så vet han att det är många som upplever att RUP är en utvecklingsmetod för IT system och det är många system som de bygger är inbyggda system, då saknar man hårdvare hantering i RUP, det gör att man vänder sig emot metoden och processen som sådan, men tycker att man i fler och fler företag säger att detta är ju de facto standard och den är tillräckligt bra och det finns ingen process som är lika stor i branschen.

Den är lite för spretig för att få kvalité stämpel i typ med ISO, mätbarheten saknas. Den börjar ju bli de facto standard. Den har inte uppnått kvalitestandars måttet.. Jag tycker att RUP som process är tillräckligt bra för det mesta, faktiskt, de gånger man misslyckas med RUP beror det på att du inte förstått det att RUP är en process där du gör an adaption för ett företag och sedan gör du ytterliggare än anpassning till ditt specifika projekt , när man hoppar över dessa transformeringar då misslyckas man. Man ska alltid lägga sig på en lagom nivå, eller på lägsta möjliga nivån för att minska overheaden.

Det är skillnad av grad av acceptans vilken roll det handlar om, både roll och personlighet, det finns några grundkoncept som jag tycker mig se att man har svårt att smälta, det har att göra med den iterativa karaktären, att du gör om saker, gång på gång, att man gör om samma aktivitet gång på gång med det ända som skiljer dem åt är att ena gången är att du får ut en draft av resultatet, och nästa gång så får du ut en bättre version, många organisationer är inte beredda att ta detta till sig, tittar man på individerna så är det mycket det här- jag skulle säga att i och med RUP är så stort i och för sig någonstans så komplext så måste du någonstans förstå processen innan du tillämpar den, du måste kunna mycket mer än bara din egen roll och detta tilltalar inte alla individer. De vill ha en guideline - hur gör jag mitt dagliga arbete.

Det är ingen de facto standard än, men det kan nog bli. Någon del i det hela är nog att det är beprövat att vattenfallsmetoden inte håller när det blir tillräckligt stort säger Jens.

Anna anser att man har ett bra stöd av att RUP har ett så pass gott namn och nästan lite hippt att använda, som gör att det blir enklare att få med folk. Ofta är det ledningen som börjar med att vi ska profilera oss att vi ska använda oss av RUP, det är nästan på väg till som med ISO standarden, Rational har verkligen lyckats med sin marknadsföring. Misslyckande att möta ledningens behov var ett exempel på kritik mot tillvägagångssättet att utveckla system i tidigare metoder.

Hur stor anser du att graden av acceptans är hos individer i projekten? Är det någon skillnad vilken roll det handlar om?

Jens tycker att RUP ger stöd i alla roller, men där den fått minst genomslag är hos konstruktörer i tekniska projekt, många gånger vill de inte ändra på sig, de är även så styrda i vilken miljö dom jobbar i t ex.. om du jobbar i ett programspråk som inte är objektorienterat så blir det lite krystat med alla dessa aktiviteter, och du måste definiera om många av artefakterna. Det är alltid svårast med programmerarna, de är lite "kan själv". Samtidigt finns det här med UML och de flesta programmerarna inser att de kan bli ännu bättre om de vidgar sina vyer mot arkitektur säger Anna.

Sammanfattningsvis så anser jag att acceptansen är hög ibland de inblandade i ett projekt, med vissa undantag exempelvis programmerarna i vissa tekniska projekt. I vissa fall är det ledningen som har högst grad av acceptans, de vill profilera sig som ett företag som använder sig av RUP, eftersom metoden fått så pass stor spridning inom branschen. Att acceptansen var så pass hög ansågs i vissa fall bero på att RUP nästan anses som de facto standard. En av de intervjuade tog upp att en orsak till att RUP är så pass accepterat är att det just är en process och inte ett flöde som användes som begrepp i tidigare metoder. Två av de intervjuade pratade i termer som hippt och hypat när det gäller att använda sig av RUP, detta kan väl snarast ses som ytterligare ett smart steg i Rationals marknadsföring.

Det krävs kunskap om processen för att du ska acceptera den, mer än bara din egen roll, och det finns personer som vill ha tydliga guidelines hur du sköter ditt dagliga arbete, detta får du inte i RUP och det tilltalar ju inte alla.

Sen tror jag att omvärldsförändringar eller upptäckandet av behov, är sådant att man måste förändra utvecklingen medan den sker, annars så får man snapshot tänkandet säger Jens.

### 7.1.5 Reliability – reliabilitet

Är processen uppbyggd så att process fel undviks eller blir funna innan de resulterar i produktfel? Vad anser respondenterna om processens tillförlitlighet?

Petter anser att tillförlitligheten är hög om man har implementerat processen på ett bra sätt, och med sina kontrollpunkter är den väldigt bra på att leda dig rätt, men en av svårigheten är att den är så pass stor, eftersom den är så stor och generell, om man inte gör anpassningar så har den definitivt väldigt lite av inbyggd processkvalité. Tillförlitligheten ökar även genom möjligheten med att skriva mer kompletta kravbeskrivningar med use-cases.

I de fall när vi verkligen har lyckats är när vi haft en eller två visionärer i projektet som säger att jag har visioner lita på mig, koncentrera inte bara dig på att följa dina guidelines utan vi ska jobba utifrån våra förutsättningar och då funkar det väldigt bra säger Petter.

Jens anser att RUP kan härleda i vilket use-case eller iteration där du inte är konsekvent mot vad du tidigare har beskrivit, på så sätt hjälper RUP dig att vara konsekvent, men har man beskrivit saker och ting fel i början så hjälper inte RUP dig med det. Den hjälper till så att sammanhanget hänger ihop, den skapar sammanhang. Alla delar du gör allting du utvecklar är kopplat på det sätt du själv har beskrivit. Sedan om det är rätt eller fel, bra eller dåligt, det svarar inte RUP på. Men den svarar på om det stämmer med de förutsättningar du har med dig, så RUP hjälper dig att vara konsekvent i det du gör.

Det här med milstolparna är ett sätt för att hitta fel så tidigt som möjligt i processen säger Anna. RUP är väldigt risk orienterad så redan i andra fasen ska man fokusera på det som man bedömer vara mest riskfyllt det som påverkar arkitekturen. Det är det som man ska bygga då, där är processen bra för att den tvingar dig att ta tag i det svåraste. Det kan däremot finnas en mental spärr att börja med det svåraste först, det kan ju även var så att det svåraste även är det mest kritiska i projektet.

Så processen är bra på det sättet, följer man den är det ingen stor risk att när man levererar beta realisen märker att det finns fel. Det är viktigt att inte slarva med milstolparna och se till att man har utbildat sin kund att förstå vad de här milstolparna innebär, annars kan det uppstå problem. Kommunikation är nog nyckelordet för att undvika problem, och att frysa kravbilder, så att kraven inte får förändras ända till slutet.

Sammanfattningsvis ansågs tillförlitligheten för processen var hög hos de jag intervjuade, RUP hjälpte till att vara konsekvent, samt att det finns en tydlig fokusering på risker tidigt i processen för att undvika produktfel. Detta är ju väldigt bra med tanke på att en av grundorsakerna till mjukvaruutvecklingens problem var misslyckande med att attackera risker i projekten.

En av de intervjuade ansåg att man måste kunna frysa kravbilder, så att kraven inte får förändras ända till slutet för att lyckas i ett projekt. Denna frysning av kravbilder måste i sådant fall vara verifierad av kund, annars är man tillbaka med samma problem som uppstod i tidigare utvecklingsmetoder, nämligen ett gap mellan utvecklare och användare, att man utvecklade någonting som användaren inte längre kände igen sina krav ifrån.

I tidigare utvecklingsmetoder fanns ett problem med oupptäckt inkonsekvens i krav, design, och implementation, detta löser RUP genom att hjälpa dig att vara konsekvent i ditt arbete, genom att kunna härleda till vilken iteration eller vilket use-case där det inte längre stämmer

överens med sammanhanget mot vad du tidigare beskrivit. Problem kan dock uppstå om du tidigare gjort fel i beskrivningarna så löser inte RUP detta åt dig, endast att du är konsekvent mot vad du tidigare har beskrivit.

#### 7.1.6 Robustness – robusthet

Jag ställde frågan om processen kunde fortsätta trots uppkomsten av oväntade problem, och Petter anser att det är viktigt att göra riktiga tilldelningar till milstolparna. Den stora vinsten är uppdelningen i arbetsflöden, att man har alla arbetsflöden öppna hela tiden, att processen till sin natur är iterativ, så har du skapat dig en mycket större stabilitet för oförutsedda händelser, detta var ett tydligt problem med tidigare tillvägagångssätt då ofta undantagshantering ignorerades.

Jens anser att RUP är väldigt hållbar eftersom man arbetar med ganska korta och målfokuserade insatser, och på så sätt så hanterar RUP oväntade problem eftersom den utgår ifrån en bild med att det här tror vi är en riskfylld sak, så till vida att den innerhåller okända faktorer, och därför ger man sig på den först, då får den ta den tid den tar, man lär sig ju mest inom det område där det är flest okända faktorer. På så sätt är den robust.

Anna sa att det inte finns någon direkt undantagshanterare i själva processen, men vid milstolparna så mäts hur långt man har kommit och man tittar på om det har kommit upp några nya risker. Väldigt ofta i projektet stannar man upp och tittar, var är vi nu, vart är vi på väg. Och där har man ju möjlighet att hitta på plan b redan till nästa iteration. För oftast gör man iterationer speciellt när man börjar närma sig slutet har man väldigt korta iterationer

Vad anser respondenterna om processens hållbarhet?

Eftersom RUP växte fram under 90-talet och är ju inget paradigmskifte, utan är bara en evolution, och mycket av det som finns i RUP är en dokumentering av det som de intervjuade ansåg att folk i branschen visste. Därför är den så hållbar som en process kan bli, den kommer ju successivt fyllas på med mer och mer kunskap, och om det inte blir någon väsentlig förändring inom systemutveckling, så är RUP hållbar väldigt långt framåt.

Det som kan vara till RUPs nackdel när man pratar om processens hållbarhet kan vara att den tenderar till att få lite elephantiasis eftersom den är så artefakt tung, väldigt mycket måste tas fram i grundutförandet.

Om RUP ska vara hållbar långt framåt så behövs nog en anpassning till hårdvaruhanteringen, eftersom allt fler system är inbyggda så skulle det komma någon förändring så ser jag dels det, och den andra är en quick RUP, RUP för små projekt, snabb RUP med mindre verktyg en minivariant. Den är så pass stor och generell så den står sig ju. Det som gäller är att man har förmåga att anpassa den efter sitt behov. Det krävs helt klart

Sammanfattningsvis verkar hållbarheten vara bra för RUP, eftersom den inte är en revolution utan gradvis vuxit fram från tre respekterade gurus inom området. Dessa gurus har satt betydelsen, samt att det är som en av de intervjuade sa mycket av det som finns i RUP är en dokumentering av det som vi i branschen visste.

Som en av respondenterna sa att två nya RUP varianter skulle vara intressant, detta är nog inte helt långt borta om RUP vill fortsätta sin spridning. Om det nu blir nya varianter eller om hårdvaruhantering läggs in i RUP det får väl framtiden påvisa. Det intressanta var förslaget med en light version av RUP för mindre projekt.

Robustheten är hög i RUP anser alla respondenter, detta på grund av RUPs fokusering på risker i projekten. Att processen till sin natur är iterativ, så skapas en mycket större stabilitet för oförutsedda händelser, eftersom konsekventa upprepningar gör att risker upptäcks relativt tidigt. En annan viktig aspekt angående processens robusthet är milstolparna , där sker mätningar, och man stannar upp och kontrollerar att man är på rätt spår, om inte så är fallet görs det annorlunda i nästa iteration.

Den stora vinsten är uppdelningen i arbetsflöden, att man har alla arbetsflöden öppna hela tiden, så att det går att gå tillbaka om det uppkommer oväntade problem, detta var ju ett av problemen med vattenfallsmodellen, att det var svårt att gå tillbaka i projekten.

### *7.1.7 Maintainability – varaktighet*

Anser respondenterna att det sker utveckling/förbättring av processen, från projekt till projekt, och i så fall hur?

Petter anser att det finns en utveckling ifrån Rationals sida, även från UML community, snart kommer nästa version av UML , där har lagts på lite mer egenskaper som har att göra med tekniska system så där finns en kontinuerlig förbättring. Internt mäts inte körprocessen rakt av utan körs på en anpassning av din företagsanpassning och att man pratar mycket om i processen att man ska återkoppla dina erfarenheter , så den är synnerligen lämplig att återföra förbättringar från projekt till projekt, att du ändrar ditt development case.

Jens säger att Rational är ganska aktiva med att lägga på de bitarna, de har bytt en del namn på diverse saker och händelser. Sedan är det en kunskapsfråga som är större, projektmedlemmarna är vana att arbeta på det här sättet. Och den biten som jag tror att alla som arbetar med RUP har är att skapa en förståelse hos kunden, vad det innebär att vara anbeställare av ett projekt som utförs med RUP.

Ja man blir duktigare och duktigare. Man växer själv. Det finns ingen kunskapshanterare i processen. Processen utvecklas också det kommer uppgraderingar varje år från Rational. Så den fortsätter ju också att utvecklas säger Anna.

Hur sker dokumentationen av processen?

Dokumentering av processen är åt det tekniska hållet, steg för steg, och ibland funkar det, men ibland kan det kännas lite träigt, i en mogen organisation är det bättre att göra en avsiktsförklaring med en aktivitet och sedan strunta i hur man gör i detalj. Se bara till att alla förstår vad det är de ska göra , så kan du strunta i steg för steg.

Det är mycket dokumentation i RUP, man rekommenderar att man ska ta fram mera- göra dina egna anpassningar och guidelines,

Om man kör RUP Online så har man ju verktyg som gör plats och detta är ett ganska effektivt sätt, vi har inte provat något annat. Och i den då skala ner och beskriva den process som gäller för projektet.

Dokumenteringen sker på alla plan. Man kan se systemet som en databas full med information och UML notationen som ger en förmåga att se informationen från väldigt många olika håll. Med olika vyer kan man titta på systemet och det är ett väldigt bra sätt att kommunicera med



kunder, användare, och programmerare osv.. Det är samma information men man kan titta på den på olika sätt.

Vem är det som är ansvarig för dokumentationen?

Det är olika ibland på projektet självt, just nu vi har en processingenjör som sköter dokumentationen, och i förra projektet var det metodavdelningen säger Petter. Egentligen vill man att det ska vara en linje funktion utanför projektet, som sköter dokumentationen för att du vill lägga det på något som är mer varaktigt än ett projekt. Egentligen vill du hantera förändringar i processen på två sätt det ena är att du gör exception request för ett enskilt projekt, - vi tänker inte följa vår företags anpassning, just i det här fallet passar den inte. I Kvalitets aspekt måste du tala om att du gör den annorlunda, men i grund och botten tycker vi att grundprocessen är mycket bättre - så att du ändrar bara för projektet, idet andra fallet så gör du en chance request till den som är ansvarig för att underhålla processen och ändrar i företagsanpassningen dvs. vi höjer vårt undantag till regel i stället. Hur man hanterar det organisatoriskt egentligen är det du behöver är en som är ansvarig för företagets anpassning och en som är ansvarig för projektets anpassning.

Jens anser att det är processingenjören, dokumentationen ska sedan revideras när man går ur en iteration

Alla roller har sitt ansvar och det är definierat vem som har ansvar för vad, så systemanalys har ansvar för vissa utpekade artefakter och arkitekten för något annat. Arbetet med dokumentationen sker kontinuerligt, man dokumenterar inte vid sidan av utan man dokumentera för att produkten ska växa fram. Tack vare att man kan generera kod och så där, och sedan kan man utifrån kod även generera dokumentation. Så det är väldigt tätt knutet där, och sedan har man de där milstolparna som säger att då ska det finnas.

Finns det något annat sätt än att dokumentera processen för att bevara den?

Du kan sätta en praxis, det handlar om kunskapen, vi pratar ju om en hantverksskicklighet - hur många use-case ska jag ha, antingen skriver du en regel eller så gör du en tio till femton - hur hittar jag rätt på det här, bygger du in detta i din organisation - ja då behöver du ju inte dokumentera det. Detta kräver att du har en kontinuerlig kunskaps hantering , och en bra kunskapsorganisation som hanterar detta.

Sammanfattningsvis så stärks varaktigheten genom att kontinuerlig förbättring sker från Rationals sida, de lägger till bitar hela tiden och uppgraderar processen efterhand, nya begrepp och händelser tillkommer till processen. I individuell aspekt så anser de intervjuade att deras kunskap ökar från projekt till projekt.

Problem med dokumentation var ett av problemen med tidigare metoder att den riktades enbart till utvecklarna och inte till användarna, detta problem verkar inte helt vara löst med RUP då dokumenteringen anses av respondenterna fortfarande som ganska teknisk.

En viktig aspekt angående varaktigheten är att hjälpa kunden att skapa förståelse för processen. En nackdel med RUP är att det inte finns någon kunskapshanterare i processen, detta anser jag kan lösas genom att man använder sig av dokumentation vid sidan av projektet för att skapa något som håller längre än själva projektet, ett problem som uppstår då är enligt en av respondenterna att dokumentering sker för att föra produkten framåt.

Två av respondenterna anser att det inte finns något annat sätt än att dokumentera processen för att bevara den, en av de intervjuade anser att man kan prova med att sätta upp praxis, hur många use-cases varje ska använda, genom exempelvis regler hur man på bästa sätt hittar rätt, om man bygger in detta i din organisation behövs igen dokumentation, men det kräver kontinuerlig kunskaps utveckling och en bra kunskapsorganisation för att hantera detta.

#### 7.1.8 Rapidity – snabbhet

Vad anser respondenterna om processens möjlighet att utveckla ett system från en specifikation?

Den är god, det är väl till och med så att du skulle kunna utveckla processen från ett business dvs. en marknad från ett kundkrav, utan en specifikation. Det som hjälper dig mest att utveckla från ett specifikation och få ett resultat som är riktigt är ju det här att man när sant iterativ i processen , så har du förstått det iterativa elementet i RUP så är det en väldigt bra process säger Petter. En sak som man har gjort bra i RUP är att man trycker på att alltid ska gå igenom alla arbetsflöden i en iteration, även om det är en undantag. Även i de tidiga faserna säger man att man ska gå ner i implementation , och följer man det , och då pratar jag om avsikten att gå ner på så kallad kodflödet det är det som stödjer dig att levererar rätt saker i slutändan. Petter tycker att just den indelningen i arbetsflöden, faser och iterationer som är en av de absolut största fördelarna med RUP.

Anna säger att det är jättebra om man vet vad man gör. Men det beror helt på vilken erfarenhet man har ifrån början. För är man erfaren och kör krav, design, kod så fungerar det väldigt bra. Men ska man under resans gång lära sig processen, lära sig verktygen ja då är den inte så snabb.

Hur ser Tidsplaneringen ut?

Återigen är det iterativa tänkandet som hjälper dig att – man är lite otydlig för man säger att man ska bara göra en fasplan men sen ska du bara detaljplanera en iteration framåt. Jag har läst den instruktionen och håller med - fast jag har sett att den feltolkas – varje projekt jag har varit med så har man feltolkat så att man sätter upp ett slutmål och gör bara en iteration, du ser inte om det pekar mot slutmålet eller inte, man har inte lyckas förmedla den intensionen att du måste ju ha några delmål också. Du kan inte bara sikta mot slutmålet.

För tidsplaneringen för ett helt projekt tror jag fortfarande lite hädiskt att det är summan av erfarenhet och fingret i luften syndromet. Har du erfarenhet ifrån projekt inom samma miljö, bransch osv.. då vet man bakrundsfaktorerna ganska bra.

När man sätter upp iterationerna blir det en tydlighet i varje leveranspunkt, det blir små vattenfalls steg där varje steg är så pass kort så att du hinner ha överblick vad som händer från a till ö.

Genom att trycka ner det till dessa korta iterationerna, då jobbar man i en ganska hård timebox – vad är det som ska levereras kopplade till use case. Då vet du vad det är för artefakter som ska komma ut, vad du ska leverera under den här tidsperioden, och på så sätt blir det hårdare styrning av att du faktiskt levererar i tid. Den andra vinningen med RUP är den delen de kallar integrationen som varje gång du levererar så integrerar du det med det föregående. Här hade vattenfallsmetoden ofta problem i slutet att de helt enkelt inte kunde integrera stora delar av systemet. Så i RUP ligger testandet utspjutt över hela tiden.

Det ser ut som översiktsskildern och sedan får man titta på hur komplext är det här problemet. Vad har vi för erfarenhet i projektet och utifrån det får man försöka göra någon vettig tidsplanering. Men jag försöker jobba så att man planerar, man estimerar den första milstolpen den kan man säga är hyfsad säker att man når dit, och de andra är uppskattningar och vid varje milstolpe så måste man faktiskt få lov att omförhandla till nästa milstolpe för man lär sig ju så mycket efter resans gång. Om problemområdet, så jag tror inte på fast pris per projekt.

Vad anser respondenterna om processens effektivitet?

Har man grupper av människor som är kunniga i ämnet så är det effektivare, men RUP har en ganska hög inestegs tröskel. Samtliga projektdeltagare släpper frågorna från gamla beprövade arbetsätt. Och då finns det alltid risker med skugg dokumentation, ja vi gör så säger man på mötet, men gör på ett annat sätt i praktiken, man levererar fortfarande samma resultat men kanske har kommit fram till det på ett sätt som inte var tänkt att man skulle jobba efter. Så för mig ställer RUP större krav på kunnigheten av processen hos samtliga deltagare än tidigare metoder. Tidigare metoder har inte lagt sig i så mycket hur du gör själv vid ditt eget skrivbord så att säga.

Har du förstått det iterativa elementet i RUP så är det en väldigt bra process. En sak som är bra i RUP är att man trycker på att alltid gå igenom alla arbetsflöden i en iteration, även om det är ett undantag. Även i de tidiga faserna säger man att man ska gå ner i implementation.

En av anledningarna till att RUP känns snabbare än tidigare utvecklingsmetoder kan vara att man hjälps till att vara konsekvent i arbetet. Har man erfarenhet från tidigare projekt så är processen snabb, men ska man under projektet lära sig förstå processen så blir den av naturliga skäl inte speciellt snabb. RUP kräver väldigt mycket dokumentation och detta borde tycker jag göra processen lite långsammare.

## 7.2 Hägerfors

**Anser du att RUP gör det möjligt att välja tekniker och metoder för att utföra arbetet så att de överensstämmer med situationen, organisationen, personer och mål.?**

Jag anser att RUP som process är så pass integrerad med sina verktyg och tekniker att det skulle var svårt att välja andra tekniker och metoder än de Rational tillhandahåller. De tekniker och metoder som Rational har fungerar bra att anpassa till rådande situation, organisation, personer och mål. De fungerar som ett ramverk som kräver att du tänker till för att anpassa till din organisation och aktuellt projekt.

En av de intervjuade tycker att man ska oavsett process eller metod kunna välja fritt tekniker och verktyg. Men ser svårigheten med att byta ut exempelvis UML som ju är en standard notation och det skulle vara svårt att ersätta denna.

Tar du bort betydande delar av RUP får du svårigheter, speciellt det med use-case hanteringen, att i stället skriva funktionella krav gör att det blir svårare. I arkitekturen har man ett visst antal beskrivningsformer som är dina typer av UML diagram, detta för att för att visualisera arkitekturen.

### **Anser du att teknikerna och metoderna används på ett kompetent sätt.? Hur?**

Metoderna och teknikerna används på ett kompetent sätt i den aspekten att de ser till att man inte löser fel problem eftersom en kontinuerlig kontroll sker gentemot användarna, att man är på rätt väg både vad det gäller risker och uppkomna problem.

En svårighet att använda dessa tekniker på ett kompetent sätt är att det fortfarande är väldigt få som kan RUP tillräckligt bra. RUP är så pass nytt att det i många fall är första projektet både för organisationen och för personerna i projektet. Ytterligare en komplikation är att det inte finns några referenser på genomförda projekt hos samma kund eller verksamhet. Detta leder i sin tur till att man har använt sig av metoden i för stor utsträckning, man inför för mycket på en gång. Det krävs nog även att organisationen anpassar sig till att arbeta efter metoden, inte bara inom projekten, om den ska kunna användas på kompetentast sätt.

### **Anser du att RUP fördelar arbetet på ett bra sätt med hänsyn till tillgängliga personer.? Hur?**

RUP fördelar arbetet bra i och med det som gäller roller, det att man frikopplar rollerna i projektet från din organisation, att den bästa personen kan vara både analytiker och testare. Med varje roll följer ett syfte i projektet, du har en metodbeskrivning, och du har ett antal aktiviteter som kopplas till varje roll. En viktig aspekt som kommer in är skickligheten hos processingenjören och projektledaren. Att bedöma vad behovet egentligen är och sedan fördela detta på rollerna.

Man har människor som går in och ur projektet på ett tydligare sätt än vad du har i andra metoder. Man utgår ifrån - vad är det för kompetens jag behöver och sedan letar man reda på den.

När man har en process som kräver att man definierar roller så kräver den ju också att vi definierar ansvar och detta kan var ganska skönt anser de intervjuade. De tycker att det kan vara skönt att luta sig mot en process, och att ansvaret är definierat på så sätt elimineras möjligheten att ansvar ramlar mellan stolarna.

### **Finns all behövlig kompetens tillgänglig ?**

De intervjuade anser att det finns ganska många som har tillräcklig kunskap om RUP, men de väldigt duktiga dvs. visionärerna saknas fortfarande. De tycker att det är rätt beslående att inte ens de riktigt stora leverantörerna har tillräcklig med folk att skicka ut, det finns helt enkelt för få. Det är brist på bra kurser som lär ut RUP. En av anledningarna till att det anses finnas för få duktiga människor med kunskap om RUP anser jag vara det faktum att du måste vara certifierad av Rational för att kunna bli rekommenderad. Så det finns kanske kompetent personal, men det svåra är att hitta dessa som inte har certifieringar hos Rational.

Sedan finns det en skillnad i kunskap mellan de olika rollerna, projektledaren och processingenjörens arbete kräver ofta mer och djupare kunskap och erfarenhet i RUP än de andra rollerna.

Det finns ju expertkonsulter från Rational att kalla in både vad gäller process och vad som gäller verktyg. Men dessa kostar väldigt mycket pengar, men det fungerar bra att kalla in dessa konsulter, speciellt i början på ett projekt och som kompetensöverföring är det ganska bra att ta in någon som hjälper en att komma i gång anser Anna.

Att vara systemarkitekt per RUPs definition kan innebära att man får ändra på de metoder eller arbetssätt man har inom sin organisation, men det är mera att lära känna RUP och begripa vad det är som jag ska leverera, än att stödja ihop processen på ett sådant sätt att den just passar ett utvecklingsprojekt. Där ligger en väldigt tung roll säger Jens.

## 7.3 Checkland & Scholes

### 7.3.1 *Efficiency*

*Resulterar processen i den önskade produkten?*

Det finns ingen generell process som hjälper dig att ta fram den önskade produkten, men RUP som iterativ process med det modellstöd som finns, och just det med att du har erfarenhets återkoppling gör dig bättre på att ta fram rätt produkt. Det blir mycket lättare att göra produkten med en bra process, och RUP är en bra process för att successivt höja kvalitén i en produkt. RUP levererar en produkt med de kraven som man var överens om, det hjälper den till att göra, sedan om detta är rätt eller fel tycker jag ligger utan för en process skyldigheter.

Det är en av farorna med Rational's eget sätt att argumentera för den, man levererar rätt sak, i rätt tid, till rätt pris i någon vid bemärkelse, och det man gör egentligen är att man sträcker sig att man utvecklat något på ett bra och konsekvent sätt.

Om processen resulterar i den önskade produkten är beroende på hur man lyckas använda processen. Det finns inget i processen enligt Rational som tyder på att det inte ska bli något annat en högkvalitativ produkt. Men det är helt beroende på situationen, vilken teknik man använder och vilken kompetens man har i projektet. Jag tycker att RUP som ramverk ger väldigt bra förutsättningar, men det är ju ingen garanti för att lyckas.

### 7.3.2 *Efficiency*

*Används minimala resurser? Finns det andra sätt, som är billigare, t ex andra tekniker?*

Det skulle till en början bli billigare att använda andra tekniker, men frågan är om det blir det i slutändan, RUP är ganska resurskrävande, men det är mycket större chans att lyckas med en metod. Alternativet att använda sig av de grundläggande delarna av RUP, och sedan använda billigare tekniker, men då missar man de stora fördelarna med Rational's egna verktyg som är integrerade i processen som leder till enklare kommunikation osv..

Väljer man att utveckla utan en metod blir det till en början billigare, men det kan i slutskedet uppkomma stora kostnader som man inte räknat med, exempel förändrade krav som gör att man måste använda sig av en annan arkitektur, man vet inte ens om man till slut hittar en lösning på problemet, man vet inte om man bygger rätt sak, inte heller om man gör det på rätt sätt. Så det är inte säkert att det blir dyrare med RUP, för det är större chans att man går i mål.

Det finns en koppling till storlek, utveckling, eller nyutveckling av något som redan finns. Är det utvecklat med RUP processen så finns det en viss typ av dokumentation som troligen gör att det går väldigt snabbt och smidigt att göra en vidareutveckling. Är projektet litet kan man kanske strunta i att arbeta efter en stor och otymplig metod. Är det utvecklar med gamla tekniker så passar kanske äldre metoder bättre osv.. RUP som metod passar ju inte överallt.

### *7.3.3 Effectivness*

*Leder metoden till uppnående av långsiktiga mål, t ex ökad marknadsandel eller god arbetsmiljö?*

De intervjuade anser att RUP ger förutsättningar för att uppnå långsiktiga mål. De anser att RUP är en bra process som gör ditt arbete lättare och du får en större förståelse varför du finns i organisationen och du kan koncentrera dig på det som är svårt. RUP löser strukturerings och planerings problemet åt dig så att du kan koncentrera dig på att bygga system, det i sin tur gör att du får en bättre arbetssituation. Problem finns så länge du inte är tillräckligt insatt i processen då projektet kan kännas utom kontroll, att man inte känner igen mycket av de traditionella styrparametrarna.

En av de intervjuade anser att det stundtals kan var hög belastning på de roller och personer som är ansvariga för projektet, men när det väl kommit igång så ger RUP en gemensam begreppsapparat som i slutändan gör ditt arbete lite enklare.

Vad det gäller långsiktiga mål beträffande ekonomisk karaktär så anser jag att en bättre arbetsmiljö leder till en nöjd och mer effektiv personal. En spekulation är att om RUP blir den standard utvecklingsmetod som flera tror, då har ju givetvis de organisationer som använder sig av metoden ett par års försprång gentemot andra konkurrenter.

### *7.3.4 Ethicality*

*Hur anser du att processen uppfattas?*

RUP uppfattas av vissa som stor och komplex. I och med att man bara har fokuserat på programvara så inom IT-branschen uppfattas den med att hänga ihop med IT system. En av de intervjuade anser att en del av problematiken ligger i hur Rational sköter presentationen– att den marknadsförs som ”allting under solen” då kan den uppfattas att lida lite av elephantiasis av många. Petter säger att kollegor till honom som kan RUP uppfattar metoden som bra, det är lite som man säger om Microsofts produkter, jag gillar inte det de står för men, men jag använder den ändå. För det är der bästa alternativet, och alla andra använder det också.

Många anser att RUP är svår att lära sig, att processen har hög ingångs tröskel. Eftersom det fortfarande är ont om referenser hos samma kund kan detta än så länge skapa osäkerhet. En anledning till att osäkerhet upplevs har att göra med att man inte har samma kontroll parametrar som man tidigare hade inbyggt.

Anna anser att den uppfattas av många som den bästa på marknaden. Både vad det gäller process och verktygsstöd. Men samtidigt finns uppfattningen att det är dyrt och det är krångligt. Det finns andra alternativ men inte av den digniteten, men sen kan man ju fråga sig om behöver en process av den digniteten. Man kan säga att om alla i teamet har lika erfarenhet av problemområdet och har arbetat tillsammans och så där, då behöver man ju ingen process. Men man har en process för att det är aldrig så att alla är lika erfarna. Man har en process för att hjälpa en framåt. Man kan ju alltid snickra ihop sin egen process i projektet men det tar ju också tid och pengar. Så därför har den säkert blivit så populär för att man kan köpa den.

Det finns företag som har en egen metod men då krävs det att det finns en metodavdelning som ständigt arbetar för att förbättra metoden, så genom att använda sig av RUP kan man säga att man outsourcar sin metodavdelning

### 7.3.5 Elegance

*Leder användandet av metoden till att verksamheten fungerar bättre och smidigare? Är processen utformad på ett tilltalande sätt?*

Det beror på hur pass mycket erfarenhet du har av metoden, fortfarande träffar vi på tillräckligt många som inte kan ta sig in i processen den har på något sätt för många vägar in säger Petter. Är man inte van vid att göra adaptationer kan det bli svårt, sen det här med att de har bilder till allt är väldigt bra. Det när man tittar på flöden, roller och artefakter, och kopplar roller, artefakter och aktiviteter till varan kors och tvärs det gillar jag, men jag tycker att man fortfarande har en viss barn sjukdom med det här med webb gränssnittet – men det är nog mera det här med det konceptuella webbhantering - hur lägger man upp en process på webben – och alla hittar rätt?

Rational har paketerat RUP som en process som tål allt ifrån att 4 personer jobbar med något i 2 månader, till att 50 personer jobbar med något hur länge som helst. Det finns för mycket, det är som vilken traditionell metod som helst att du får fyra pärmar i handen, och då är det ingen som orkar arbeta efter den. RUP har lite samma syndrom, även om det ligger i maskinen så är det en fruktansvärd mängd information. Det är användaren som måste lära sig att skala. För att kunna detta krävs kunskaper och erfarenhet.

## 7.4 Sammanfattning

*För att få en överblick av analysen och få svar på de frågeställningar denna uppsats bygger på svarar jag sist i denna analys på frågorna ifrån inledningen.*

### **Leder användningen av RUP till en förbättrad systemutvecklingsprocess?**

RUP har alla förutsättningar för att leda till en förbättrad systemutvecklingsprocess. RUP är väl definierad och det är relativt lätt att förstå processen, men det krävs kunskap för att kunna plocka ut det relevanta. Eftersom RUP börjar bli de facto standard kan du börja relatera till samma metod och detta ökar kundförståelsen. Kommunikationen mellan de inblandade i ett projekt underlättas genom UML, att alla pratar samma ”språk”.

Varje aktivitet skapar en artefakt och RUP har sina milstolpar och uppdelning i iterationer, detta gör det möjligt för kunden att se vad som sker inne i processen. Det finns en röd tråd genom varje projekt och det är processöversikten där ser man var man är, samt vart man är på väg. Det finns dock en skillnad i synlighet för den enskilde inom ett projekt, och synligheten gentemot kund.

Jag anser att det finns ett verktyg för allt i RUP, alla aktiviteter kan stödjas på ett eller annat sätt. Användningen beror lite på hur pass mogen utvecklingsorganisation är. Det finns ett komplett utbud med tillräckligt bra verktyg. Man behöver inte köpa hela konceptet med verktyg utan anpassar det efter projektets behov eftersom det är väldigt dyrt.

Acceptansen är hög bland de inblandade i ett projekt, speciellt ledningen verkar vilja profilera sig som ett företag som använder RUP. RUP hjälper dig att vara konsekvent, och att upptäcka risker tidigt i processen detta gör att tillförlitligheten för processen är hög.

Eftersom RUP inte är någon revolution utan har gradvis vuxit fram ifrån tre respekterade gurus anser jag att hållbarheten av processen är lång.

Jag anser att det sker en kontinuerlig förbättring av processen ifrån Rationals sida, de lägger till eller förbättrar processen efterhand.

Processen möjlighet att utveckla ett system från en specifikation är god, det är till och med möjligt att utveckla ett system utifrån ett kundkrav utan en specifikation.

En av anledningarna till att RUP känns snabbare än tidigare utvecklingsmetoder är att du hjälps till att vara konsekvent i arbetet. Har man erfarenhet från tidigare projekt så är processen snabb, men ska man under projektet lära sig förstå processen så blir den av naturliga skäl inte speciellt snabb. RUP ställer större krav på kunskap av processen hos samtliga deltagare än tidigare metoder.

### **Vilka är fördelarna/nackdelarna med denna metod?**

Att allt är paketerat och samlat på ett ställe är en fördel, men kan även bli en nackdel då det kan bli för mycket information och svårt att hitta det väsentliga, så för att det ska vara en fördel måste man lära sig plocka ut det relevanta.

Att RUP snart är de facto standard är naturligtvis en fördel för de organisationer som används sig av metoden en tid. Ytterligare en fördel med RUP är att den underlättar kommunikationen mellan de inblandade i ett projekt genom användningen av UML, även kommunikationen gentemot kund underlättas om kunden är insatt i UML.

En av nackdelarna med denna metod är att den kan uppfattas som stor och komplex. Ytterligare en nackdel är att det finns en ingångströskel innan du kan använda metoden effektivt och fullt ut. Priset är även det en nackdel, och du måste plocka ut det som är relevant för respektive projekt.

### **Är kritiken befogad mot det traditionella tillvägagångssättet, är RUP lösningen på kritiken?**

Mycket av kritiken mot det traditionella sättet att utveckla system är faktiskt RUP lösningen på. Att du tidigt i processen hanterar risker och att utvecklingen sker iterativt, samt kontinuerlig dokumentation leder till större chans att lyckas med ditt projekt.

En förutsättning för att RUP ska vara lösningen på dessa problem ifrån tidigare metoder är att du är väl utbildad och har stor erfarenhet av att arbeta efter RUP. Saknar du denna erfarenhet och utbildning så försvinner fördelarna med metoden och den känns bara stor och komplex. Så det är en inlärningströskel innan metoden löser dessa problem ifrån tidigare metoder.



## 8 Diskussion

Vad som framkommit i denna studie är att RUP leder till en förbättrad systemutvecklingsprocess, och att metoden faktiskt hanterar många av de problem som tidigare metoder har haft. RUP förutsätter däremot en stor kunskap och erfarenhet om metoden och dess begrepp, för att kunna använda metoden effektivt.

Det som respondenterna var kritiska till är det faktum att du måste vara licensierad av Rational för att kunna bli rekommenderad som användare av RUP. Det var svårt att hitta kritik till själva metoden, respondenterna var ofta mer kritiska till hur företaget Rational sköter sig än till själva metoden.

Vidare märkte jag att skaparna av RUP var väldigt respekterade, att de hade skapat en ny metod som löste problemen ifrån tidigare metoder. Vad som var tydligt var det faktum att UML har haft en stor betydelse för acceptansen för RUP, eftersom denna notation har använts länge och av många inom branschen.

Jag och min handledare valde att lägga resultatet ifrån intervjuerna i bilagan för att det annars bitvis kunde kännas som upprening om vad respondenterna svarade.

Kritik till eget arbete var att jag inte insåg att det krävdes djupa kunskaper i RUP för att kunna svara på mina frågor, och detta ledde i sin tur till svårigheter att hitta lämpliga respondenter. Vad det gäller uppsatsen reliabilitet och validitet kan anmärkas att respondenterna kanske inte i alla fall varit helt objektiva eftersom de själva utbildat sig i metoden och därför kan ha varit lite partiska.

En annan aspekt vad det gäller framtiden så har jag precis läst att IBM köpt Rational, och därmed även RUP, intressant att se vad det får för konsekvenser i framtiden.

## 9 Framtida forskning

Under arbetet med min uppsats har det dykt upp flera intressanta frågor som jag tyvärr var tvungen att välja bort. Det skulle vara intressant att undersöka RUPs effektivitet i små respektive stora projekt, vid olika storlek på organisation osv.. Det skulle var väldigt intressant att göra en undersökning om ett par år och se utifall RUP har blivit standardmetoden för systemutveckling och vad det har fått för konsekvenser på systemutvecklingen.

## 10 REFERENSER

### Litteratur

- Andersson, H. 1994 *Vetenskapsteori och metodlära*, Lund: Studentlitteratur.
- Avison & Fitzgerald. 1997. *Information Systems Development – Methodologies, Techniques and tools 2<sup>nd</sup> edition*. London, Storbritanien: McGraw-Hill.
- Bergman & Klefsjö. 1995. *Kvalitet från behov till användning*. Lund: Studentlitteratur.
- Checkland & Scholes. 1990. *Soft system methodology in action*. Chichester : Wiley.
- Holme & Solvang, 1997 *Forskningsmetodik*, Studentlitteratur
- Hägerfors, Ann. 1995. *Att samlära i systemdesign*. Lund: Studentlitteratur.
- Karlsson & Söderstedt. 1997. *ISO 9000 – Från teori till praktik*. Uddevalla: Bonnier DataMedia.
- Kruchten, Philippe. 2000. *The Rational Unified Process – An introduction 2<sup>nd</sup> edition*. Reading, USA: Addison Wesley Longman Inc.
- Lundahl & Skärvad, 1992 *Utredningsmetodik för samhällsvetare och ekonomer*, Lund: Studentlitteratur.
- Patel & Davidsson. 1993. *Forskningsmetodikens grunder*, Lund: Studentlitteratur.
- Pressman, Roger S. 2001. *Software Engineering – A Practitioner's approach*. New York, USA: McGraw-Hill.
- Sommerville, Ian. 2001. *Software Engineering*. Harlow, Storbritanien: Addison Wesley Ltd.
- Thurén, 1991. *Vetenskapsteori för nybörjare*, Liber,
- Wallén, 1996 *Vetenskapsteori och forskningsmetodik*, Studentlitteratur.
- Wiedersheim-Paul & Eriksson, 1993. *Att utreda, forska och rapportera*, Hermods: Liber

## APPENDIX MED INTERVJUERNA

### A.1 Sommerville

#### A.1.1 Understandability – förståelighet

**Till vilken grad är processen definierad och hur enkelt är det att förstå process definitionen?**

##### **Petter**

RUP är väl definierad och kanske i vissa fall nästan över definierad. RUP har flera ingångar utifrån det arbete som ska göras - aktiviteterna, workflows, ordningen som ska följas, iterationer, artefakter och roller alla de behövs men bekymmer blir att det skapas så många vägar in. Blir svårt att förstå just därför du har så många vägar in till detaljerna så att du inte vet var du har hamnat.

##### **Jens**

Jag tycker att den är väldigt enkel egentligen, att den är självklar. Grundidéerna har man jobbat med traditionella projekt med vattenfallsmodellen så är punkterna där RUP fokuserar ganska självklara. Det är inget revolutionerande, ingen frälsning, de har paketerat ett gäng välkända faktorer på ett bra sätt. Och att man har adresserat om ordningen hur man angriper saker är det stora.

##### **Anna**

Den är väldigt väl dokumenterad, problemet kan vara att den är så pass stor, så det kräver erfarenhet för att kunna veta vad man ska plocka, för det är meningen att man ska anpassa processen till sitt projekt. Det talar Rational om tydligt. RUP är väl dokumenterad och man kan ta den till sig själv genom att köpa licenser, men ska det gå bra ska man först gå en kurs i RUP och sedan jobba i projekt med RUP.

**Hur tycker du förståelsen av process begreppen är i RUP? Ex terminologi?**

**Är det många återkommande begrepp ifrån tidigare metoder, har i så fall dessa begrepp gjort så att det blivit förbättringar/förenklningar?**

##### **Petter**

Det som är tydligt i RUP är att begreppen kommer ifrån Jacobson, Rumbaugh och Booch, de har vävt ihop sina begreppsvärldar och i många fall tycker jag om man tittar på use-cases så är det ju Jacobson som därför har han satt betydelsen. Svårigheten är den att du måste inte bara veta vad de säger utan även vad de avser, den sk. intent vad var avsikten, det gör att man inte kan tro att du kan läsa RUP rak upp och ner och tro att ska förstå, det är som vi brukar säga att du först måste läsa helheten en gång för att kunna sätta dig in i det. I vissa projekt används guidelines, och kan du inte begreppen hjälper de inte till, pga. detaljeringsnivån, och förstår du så behövs de inte, för då känns de lite stela. När du väl förstår RUP är det mycket bra, och det är nog tack vare skaparnas långa erfarenhet och att de har hittat begrepp som saknads tidigare.

Ofta pratar vi om att du måste skradda sy allt, så att du gör det begripligt för din publik och ibland handlar detta om eftersom det finns flera vägar in, att bara ta bort information så att det bara finns en väg in, du måste alltid utgå ifrån aktiviteterna.

De facto standard jo då vävt ihop fler kulturer generellt ökad komplexitet pga. av varje ny begreppsvärld vi lägger på. ex i ett projekt jag arbetar med nu har vi lagt på militär standarden 498 och dess begrepps värld, och varje gång du lägger till något ökar komplexiteten och jag skulle snarare säga att kan du och har erfarenhet av RUP så visst är det bättre för du får en mycket mer komplett palett att ta av. Varje enskild komponent kan du hitta någon annan stans.

### **Jens**

Läser man förhållandevis mycket engelsk och amerikansk litteratur i ämnet och driver projekt på engelska och där koncernspråket är engelska, så är det inte speciellt jobbigt. Men företag som har en svenskprofil och vill ha RUP på svenska, de som man kan se är svårigheten att skilja på person och roll, att samma person kan ha flera roller. Mottagaren av projekt har svårast att hantera detta.

Jag tycker inte att det är något nytt med RUP, utan det är sättet och ordningen att göra det som är nytt ex. det här med riskadresseringen. Vi på Cap jobbar med IAD –iterativ applikation development det är en metod med timeboxar på 10 veckors perioder, som ett rullande schema. Det är samma grundide´ så här på Cap är RUP inte ett nytt sätt att tänka. Man är van med iterationer och man är van med korta tidsrymder.

### **Anna**

Ja det tycker jag för den bygger på UML, där har RUP ett försprång gentemot många processer att det är en standard notation som man redan kan om man sysslar med utveckling i branschen, så den är lätt att förstå.

Det hade nog inte vart någonting om det inte varit UML, för egentligen inget revolutionerande med RUP för det är ju vanligt folkvett, men just det att det är paketerat, det är det som är grejen.

### **Tycker du att man får ökad kundförståelse med hjälp av att använda RUP?**

#### **Petter**

Ja i två aspekter ökar kundförståelsen att RUP börjar bli de facto standard så att du kan börja relatera till samma metod. Tidigare har det ofta varit så att du har haft en metod och kunden en annan. RUP gör väldigt bra ex att de lägger på detta business workflow och den hanteringen, och hur man lyfter fram olika modeller för att kunna kommunicera med kunderna. Det jag sett successivt är att gör du dina användningsfall rätt ger det väldigt mycket bättre kundkontakt.

I de implementationer vi har gjort använder vi use-cases i kravflödet för att krav av en viss typ. Försöker göra en bild av systemets beteende. Att alltid hela tiden sikta på att få beskrivningen så komplett och begriplig så att du kan ta den till kund och test och verifiering. Och säga så här kommer systemet fungera, alltså redan från dag ett funderar och resonera man på hur man kommer att uppleva systemet, inte hur det kommer att var byggt utan man hoppar över den funktionella aspekten. Återigen detta med bilder så använder sig RUP av UML diagram, och när organisationen lärt sig att kommunicera genom dessa får du mycket bättre förståelse.

#### **Jens**

Förståelsen från kund då är det tvärtom, det är mycket sämre med RUP än någon traditionell metod. Om man sedan ser på kundens verksamhet, eller vad det är som ska lösas så tror jag att man når en högre detaljnivå fortare inom riskområdena med hjälp av RUP. Där tror jag att de

stora skillnaderna ligger, att man tydligt adresserar att varje iteration löper ut i något som är en exekverbar sak. Om det sedan är på papper eller en liten pilot eller vad det nu är så kommer man ganska djupt ner i detaljer ibland, och därför kan man snabbare nå förståelse för kundens verksamhet inom vissa områden, det följer de use cases eller de områden där du fokuserar.

Det jag tycker är väsentligt är att det finns ett bra underbyggt business case, man kollar av vad är det för process vi ska göra något i, hur ska den stödja den så man kommer in med use case arbetet på rätt sätt, då kan man möjligen kartlägga var och hur saker fungerar och hur man kanske vill att det ska fungera, men en kund måste alltid vara aktiv i processen och tala om vad han vill ha. Om han inte vet måste han definitivt - om han inte vet, vara väldigt aktiv och ta reda på vad han vill i processen.

### **Anna**

Ja det är ju olika arbetsflöden i RUP där requirement är helt fokuserad på att förstå vad kunden vill ha, väldigt fokuserad på att hjälpa kunden att veta vad den vill ha. Den är jättebra det arbetsflödet. Det blir alltså åt båda håll. Så även om man inte vill arbeta enligt hela RUP kan man plocka åt sig väldigt mycket bra ifrån t ex. requirement.

### **Underlättas kommunikationen mellan de inblandade i ett projekt? Hur?**

#### **Petter**

Kommunikationen underlättas inom projekten tack vare UML, stödet för det iterativa arbetssättet med arbetsflöden och med vissa roller, som ex arbetsflödet analys och design använder sig av roller i stället för en organisatorisk enhet, det kan ju vara samma person genom det har underlättat uppfattningen vad som behöver göras, och när, du har alltså frikopplat de olika arbetsuppgifterna från varandra. Ex RM flödet är ju aktivt från dag ett, och alla andra flöden är ju också aktiva hela tiden. Det finns element dels det med syntaxen och de gemensamma beskrivningsformerna av alla arbetsuppgifter som är rätt bra, bara det att RUP är så pass komplett underlättar.

#### **Jens**

Det finns en tydlighet i artefakterna som gör det lättare att kommunicera. Vissa tekniker och notationer är gemensamma, UML, use case beskrivningarna. Det finns en tydlighet i vad man pratar om som kan vara en fördel, var man befinner sig i utvecklandet eller konstruktionen av olika saker. Där tänker man lite annorlunda eftersom man kopplar mot use casen hela tiden, det finns en tydligare fokusering.

#### **Anna**

Ja det är sagt att det ska underlättas i alla fall, eftersom alla i projektet ska prata UML, och de olika verktygen som är kopplade till processen, det är ju valfritt att använda dessa men, där är det tänkt att både kravmänniskorna och de som designar gränssnitt och de som designar databaser ska använda sig av UML, och så gärna samma verktyg ex. Rational Rose så processen ger ett väldigt bra stöd för den kommunikationen. Sen om man klarat det fullt ut är ju en annan sak, men stödet finns där, och målsättningen med RUP är ju just det att man ska underlätta kommunikation. Annars är det väldigt lätt hänt att det blir tre läger. Du har kravgänget på sin flank och sedan de som programmerar och sedan databas människorna på sin flank som optimerar och fixar till databasen. Det är ju inte säkert att dessa tre håller ihop. För att det ska fungera bra ska alla dessa tre tycka att det är roligt och intressant med UML och objektorientering.

## A.1.2 Visualibility – synlighet

**Ger process aktiviteterna klara resultat så att framåtskridandet av processen externt blir synliga?**

### **Petter**

Ja och nej genom detta med artefakt tänkandet att varje aktivitet producerar en artefakt du har dina milstenar och uppdelning i interaktion, men du har även separata aktiviteter – du ska göra det här och det här- resultatet blir deployment view i analysmodellen. Där tycker jag att man är bra framme, men det man missar i RUP är att det kan finnas en nackdel med att ha en process som är så pass komplett. Du leds till att tro att den är mer precis. Att det finns mer i den, det kan man se aktiviteterna har ju tydliga resultat som ex vilket dokument som ska produceras du måste själv tala om hur mycket du ska producera, du ska ex lägga ut en draft av en modell ..vad är en draft ..du har ingen känsla för vad en draft är det hälften eller alla klasser? Där är man sämre tycker jag.

### **Jens**

Om man börjar uppifrån i en kundorganisation vill de alltid veta vad det kommer att kosta, hur mycket folk behövs, och hur lång tid tar det, och när levererar du vad? Ett riskorienterat projekt med ganska mycket okända faktorer kommer ju inte att leverera speciellt mycket under den första tiden, och det gör ju att projektet snarare blir mer osynligt, i form av att det inte fysiskt visar vad som gjorts. Detta skapar en oro över att man inte har kontroll över projektet i kundledet.

### **Anna**

Ja det måste, ja det är ju det krav man har på processen när man arbetar enligt den så det är därför man anpassar processen efter sitt projekt så att man inte bara sitter och dokumenterar och det inte blir något av det, det finns en jätte risk med RUP att man bara dokumenterar och inte bygger, så det krävs att man antingen har ett bra metodstöd eller en duktig projektledare som kan plocka det som vi verkligen ska göra i projektet. Och gör man det är svaret ja på frågan.

Även om man inte har egna milstenar i projektet så har ofta kunden det. Det är alltid en leverans som man definierar från början – vad ska vi leverera vid första milstolpen? Det är alltid kravbilderna som ska vara överenskommet vid första milstolpen. Andra milstolpen ska man komma överens om ett lösningsförslag, så då är leveransen lösningsförslaget. Tredje milstolpen så är det kärnfunktionaliteten implementerad. Fjärde är hela systemet. Där är RUP tydliga att det alltid ska vara en fysisk leverans. Så det är bra.

**Finns det en röd tråd som är synlig genom hela projektet?**

### **Petter**

Den röda tråden som finns är ju dina arbetsflöden och den s k sega råttan bilden, där du talar om vad varje workflow ska producera och någon stans vid varje milstone så har du ju ett exit kriterium. Har jag uppfyllt det, de blir den röda tråden när det gäller progress det som jag tycker är har sett när man ska planera RUP projekt att vissa har problem att se vilken aktivitet som ska göras först och vilka som kan göras parallellt. Jag har inte själv haft den uppfattningen att det är svårt att se en rödtråd i RUP, men jag har sett i ett par projekt att man har väldigt svårt för att tolka RUP, så att det kan tyda på att, även om jag ser en rödtråd så kan den vara svårt att ta till sig.

## **Jens**

Jag tycker att RUP har en bra utgångspunkt i risklistan, det är den som är kritisk och styrande för mig. I vilken ordning angriper vi saker? Det gör att vi bygger upp ex som det är med prefab hus, man gör ritningar , bygger moduler, men du ser ju inget på tomten där huset ska stå. Sedan inom loppet av en vecka blir det färdigt. När det väl blir något går det väldigt fort , men det är ganska mycket osynligt förarbete. Man jobbar ju emot sina milstones och det finns ju tydliga kriterier men det är ju en skillnad om dessa kriterier är tydliga för en beställare som kanske inte har samma insikt i sina system eller processen, och kanske ganska okunnig i RUP, då vet kunden ganska lite vad det är han får i en milstone. Detta är RUP inte speciellt duktig på att hantera, utan förutsätter att mottagaren är kunnig. RUP förutsätter nästan att processen är förankrad i kundens verksamhet.

## **Anna**

Ja processöversikten, den kartan jobbar alla efter , den planerar och följer man upp projektet efter. De fyra faserna och kärnarbetsflödena, och så milstolparna. Så som jag gör när jag planerar upp projektet börjar jag att titta över tids perspektivet för det får man nästan från början, typ vecka 40 ska det var klart. Krav arbetet tar så långtid och sedan lösningsförslaget och sedan en beta realise osv.. Arbetsflödena studeras sedan vilken typ av människor kan jag behöva ha och när, i de första veckorna behövs kanske inga stjärnprogrammerare, osv.. Jag planerar upp projektet efter det och alla som är med i projektet ska känna varm för den här modellen, för det är på så sätt vi ser att vi rör oss framåt. Och även mot kunden så kommunicerar vi status utifrån detta. I varje fas sätter man sina iterationer, ex i den första fasen har man bara en iteration, men i den andra har man två , och i den tredje har man ganska många för där släpper man ju små subsystem hela tiden så det är basen för hela , det är en bild från projektet ifrån start till mål. Det är jätte bra i RUP , översikten , alla ser var man är, jobbar man dessutom i licensen i själva RUP då kan man gå ner i de här olika arbetsflödena och hitta guidelines och mallar för de olika faserna, och det är ju jättebra men det kräver ett metodstöd eller att projektledaren är kunnig, för att som utvecklare orkar du inte ta tag i det för att börjar man gräva sig ner här så bara expanderar. Guidelines är bra, för som vi har gjort när man har planerat upp projektet så tänker man vad vi ska leverera till varje milstolpe, och utifrån det ser man vilka aktiviteter behöver vi utföra, och då kan man hitta guidelines, framför allt kan man hitta toolmentors som visar hur man gör. i tydligt. för att få fram just den artefakten, och det är extremt handfast hur man ska gå till väga. Bara titta på arkitektur dokumentation är en sak som ska komma ur milstolpe två , då kan man hitta precis hur man gör i Rose för att få fram denna arkitektur dokumentation. Det är ju otroligt bra hjälp om man är en duktig arkitekt men inte har arbetat i Rose tidigare. Men du kan inte som rudis komma och köra i gång licensen, för då är det för stort man kan verkligen gå vilse i RUP cyberspace den otroligt stor. Jag brukar göra en egen översikt bild där jag skriver de roller jag ska ha med för det finns tror jag 167 olika roller definierade i RUP. Så det första jag gör är att plocka ut de roller jag kommer att behöva. Jag placerar ut dem och sätter namn på dem, och då har man nischat in sig, varje roll har ju då ett antal aktiviteter knutna till sig så.

## **Kan kunden påverka processen under utvecklingen? Hur?**

### **Petter**

Nej det kan de ju naturligtvis men jag vet ingen organisation som skulle klara av att hantera detta , du kan som kund påverka milstenarna men inte själva processen och det kan du ju göra och det är ju en av de stora vinsterna med RUP, att alla flöden är aktiva hela tiden, så att det är ju sant iterativt , och att man dessutom har flera iterationer, så det kan du ju. RUP är bra på detta just därför att den i sin beskrivning är så tydligt iterativ.

## **Jens**

Det finns stakeholder needs och de är ju de som kunden kan adressera och förfinas hela tiden. Kunden bör vara väldigt aktiv i beskrivningen av use casen så att det verkligen svarar upp emot vad han förväntar sig. De review processerna, de genomgångarna, de förankringarna på den biten är ju oerhört viktiga. I de planerande ledet i use casen, måste kunden vara väldigt aktiv, man måste hjälpa honom att vara aktiv och hjälpa honom att förstå vilket krav han ställer. Men jag tycker att det finns en tydlighet i att han kan påverka krav under resans gång.

## **Anna**

Det beror på vad vi menar med processen, kunden påverkar projektet eftersom minst vid varje milstolpe har vi en avstämning med kunden. Där vi signar att vi är överens att gå vidare. Att kontrollera att vi uppnått de saker vi sa att vi skulle ha uppnått. Det ger ju kunden möjlighet att påverka, sedan har kunden möjlighet att påverka processen i projektet genom att det som vi ska göra i de olika iterationerna det bygger på kundens prioriteringar. Man delar upp systemet i aktörer och användningsfall. Man bygger en bild av systemet, aktörer det är de personer eller system som kommer att interagera med vårt system. När vi gjort det har vi fått en avgränsning, vad vårt system är och inte är. Vad är det de vill göra i och med systemet, det blir ett antal användningsfall, ett användningsfall är inte en funktion i systemet utan består av många funktioner. Men något som en aktör vill kunna göra i systemet. Allt arbete utgår sedan från denna användningsfalls modellen för att till varje iteration ska man ha med ett antal användningsfall, för att varje iteration ska resultera i en leverans. Det är kunden som berättar ex. till denna iteration vill jag ha med dessa tre användningsfallen. Där styr kunden helt vad vi gör i projektet, sen måste vi hjälpa kunden så att inte till första iterationen inte gör en liten skrivarrutin eller så. Först ska man göra det som är mest riskfyllt, eller det som är kärnfunktionalitet.

### A.1.3 Supportability – stödjande

#### **Till vilken grad kan process aktiviteterna stödjas med tekniker och verktyg?**

## **Petter**

Process aktiviteterna stödjas med tekniker och verktyg det är här det märks att Rational bara är en verktygs leverantör, som har tagit sig ett mycket större scope, för allt i RUP så finns det ett Rational verktyg, på ett eller annat sätt, och jag tror det har mer att göra med som Rational som företag, ja de kan stödjas, teknikerna är ju, ex vill du göra ett diagram så har du beskrivning i UML, du behöver inte köpa hela konceptet, i det projekt jag arbetar nu har man bytt ut kravhanteringsverktyget mot ett annat, alla aktiviteter kan stödjas på ett eller annat sätt, det beror lite på hur mogen utvecklingsorganisation man är, för är du en mogen utvecklingsorganisation så vill du nog ta klivet över i modellbaserat system utveckling, att den inte är text eller dokumentbaserad, och då är det ett måste att du har verktyg, då räcker kanske inte vanliga Rational Rose, utan du måste ha andra ex repository, Jag tycker inte Rational's verktyg är exceptionellt bra, men eftersom vi övat i andra projekt så kan man säga att de inte är väsentligt mycket sämre än något annat, de dras med samma problem som alla verktyg, med samma inriktning, sen kan man ibland skulle man vilja plocka ihop delar av olika verktyg, Rational har gjort rätt för de har ett komplett utbud som är tillräckligt bra, och sedan är frågan hur man hanterar en kund när han vill byta ut en komponent i deras utbud, det är väl en annan fråga i o för sig,



### **Jens**

Det förutsätter faktiskt att man kör Rationals egen leverans av RUP. Sen är den till stora delar även konstruerad för att man ska använda deras verktyg, det är deras affärsidé. Man ska köpa till Robot och Clearcase osv.. De integrerar dessa verktyg mot RUP, de har lagt på en massa, och detta tvingar ju nästan utvecklingsföretag att använda dessa. Det är inte helt trivialt att byta ut dessa verktyg, notationsteknik är ju inte att tänka på att byta ut. Sen blir det kundberoende – vad har de för versionshantering – hur vill de att vi ska dokumentera sekvensdiagram osv..

### **Anna**

I väldigt hög grad eftersom Rational inte bara levererar en process utan också verktyg. Så till och med lite väl hög grad ibland.

### **Anser du att processen blir bättre med hjälp av verktygen?**

#### **Petter**

Jag tycker inte att processen blir bättre, men processen är vad den är - frågan är om implementationen av processen blir lättare, med verktyg, verktygen underlättar dig att samla ihop resultatet, jag har varit i projekt där man dokumenterade allt i visio och det var mycket jobbigare än om du använder verktyg som Rose, modellhanterande verktyg, och sedan..du måste ju ha en kravhantering någon stans, det är ju alltid bra om du köper verktyg som är integrerade, om du köper RUP som process och dess webbapplikation och sedan dess verktyg, så naturligtvis du får en kontakt så vänd dig då till dem, så förväntar man sig att de löser dina problem.

#### **Jens**

Nej det tycker inte jag, RUP är ett sätt att förhålla sig till utvecklingen, och ett sätt att arbeta med utvecklingen och sedan vilka verktyg vi använder gör det inte till eller från, det kan göra det lite mer eller mindre jobbigt för någon att lära sig verktygen. Jag tycker att det är ett krav på en bra utvecklingsprocess att man ska kunna välja verktyg självständig, det är inte det en utvecklingsprocess säljer, alltså ett behov av ett verktyg, utan ett säkerhetsställande att jag utvecklar något på ett bra sätt, med kvalitet i varje led där man jobbar. Enkelt uttryckt ska processen säkerhetsställa att man är konsekvent från början till slut, sen om det blir en bra eller dålig produkt är en annan sak. Vi använder vanligtvis andra versionshanterare.

#### **Anna**

Ja om det är hyfsad storlek på projektet, så blir den det. Men man måste var medveten om att för varje nytt verktyg man tar in så blir det en risk också, dels så får man en massa overhead utbildning och det kan faktiskt påverka om man huruvida kommer fram till något i slutändan eller inte. Det är ganska om, om man t ex. tittar på modelleringsverktyget Rose det är väldigt centralt i RUP att man använder sig av ett modelleringsverktyg, och att man genererar kod och man är väldigt UML fokuserad. Använder man sig inte av Rose då betyder detta nästan att man stannar på någon slags projektlednings perspektiv, att man använder RUP som en projektlednings metod. Men sen finns det andra verktyg ex Clearcase som används för att hantera versioner och konfigurationer. Där kan man lika gärna ta ett annat verktyg. Det kostar väldigt mycket att använda Rationals verktyg. Nästan alla företag har ett eget projektledar verktyg, eftersom det är bransch beroende. Ex PCM, SMASH eller PROPS. Och så här blir det nästan alltid om det inte är mjukvara man producerar.

### **Dessa verktyg är de lätta / svåra verktyg att lära sig, att använda?**

### **Petter**

Processen som sådan blir nog inte så mycket bättre pga. verktygen, men däremot blir det lättare att arbeta i processen. Alla verktygen har en inbyggd svårighet, och det är svårt att arbeta med verktyg , och det finns inget verktyg i dag som har till betydande del sänkt den tröskeln.

### **Jens**

### **Anna**

De är väldigt supporterande i alla fall, det finns både kurser och expert konsulter och bra dokumentation så alla förutsättningar finns i alla fall. Sen är de väldigt sofistikerade så det finns ju en tröskel, helt klart. Det är verktyg som är fokuserade på process så det behövs fortfarande en utvecklingsmiljö där du editerar , komprimerar, och debuggar det finns inte i någon av dessa verktyg. Du behöver även ett databas hanterings system där du har din databas och bygger din modell så de verktygen är inte allt du behöver i ett projekt utan allt du behöver i en process. Till och med mer än vad man behöver i en process.

## A.1.4 Acceptability – pålitlig

### **Är den definierade processen accepterad av, och användbar för utvecklarna? Processen accepterad av ledningen? Kunderna? Produktionsansvariga?**

### **Petter**

Det beror lite på organisationen, det är en mognadsfråga, i tekniska projekt så vet jag att det är många som upplever att RUP är en utvecklingsmetod för IT system och det är många system som vi bygger är inbyggda system så saknar man hårdvaru hantering i RUP, det gör att man vänder sig emot metoden och processen som sådan, men jag tycker att man i fler och fler företag säger att detta är ju de facto standard och den är tillräckligt bra och det finns ingen process som är lika stor i branschen.

Den är lite för spretig för att få kvalité stämpel i typ med ISO, mätbarheten saknas.. den börjar ju bli de facto standard. Den har inte uppnått kvalitestandars måttet.. Jag tycker att RUP som process är tillräckligt bra för det mesta, faktiskt, de gånger man misslyckas med RUP beror det på att du inte förstått det att RUP är en process där du gör an adaption för ett företag och sedan gör du ytterliggare än anpassning till ditt specifika projekt , när man hoppar över dessa transformerings då misslyckas man. Man ska alltid lägga sig på en lagom nivå, eller på lägsta möjliga nivån för att minska overheaden..

Det är skillnad av grad av acceptanse vilken roll det handlar om , både roll och personlighet, det finns några grund koncept som jag tycker mig se att man har svårt att smälta, det har att göra med den iterativa karaktären, att du gör om saker, gång på gång , att man gör om samma aktivitet gång på gång med det ända som skiljer dem åt är att ena gången är att du får ut en draft av resultatet, och nästa gång så får du ut en bättre version , många organisationer är inte beredda att ta detta till sig , tittar man på individerna så är det mycket det här- jag skulle säga att i och med RUP är så stort i och för sig någonstans så komplext så måste du någon stans förstå processen innan du tillämpar den, du måste kunna mycket mer än bara din egen roll och detta tilltalar inte alla individer. De vill ha en guideline- hur gör jag mitt dagliga arbete.

### **Jens**

Den har ett ganska farligt snitt egentligen, mellan den rollen som kallas processingenjör och projektledare. En kritisk framgångsfaktor är att få projektledaren att ta hänsyn till vad processingenjören gör och levererar, och tvärtom. Jobbar inte dessa roller ihop så har du en roll som producerar uppföljning på ett eller annat sätt, och en roll som hamnar i ett maktvakuum som egentligen styr och leder projektet utan mandat. Det är processingenjören som säkerhetsställer att vi arbetar metodiskt. Där kan man se att det spricker.

Vill våra kunder ha RUP som process måste vi var duktiga på det. RUP är lite hypat på marknaden när det gäller att starta nya grejer. Steget till RUP känns inte så annorlunda för oss eftersom vi arbetat efter liknade metoder tidigare. Det är ingen de facto standard än, men det kan nog bli, men någon del i det hela är nog att det är beprövat att vattenfallsmetoden inte håller när det blir tillräckligt stort. Det är ganska väl dokumenterat att projekt i dag är ganska korta, ner till ca sex månader, och då är det förhållandevis ganska mycket folk inblandade. Ordet process i sig är ganska häftigt, man pratar om att man har processer i företagen, tillverkningsprocess osv.. allt är ju processer idag. För några år sedan var det flöden. Sedan är det med triangeln med tid pris och kvalité, vilket du prioriterar. Trycker du på kvalité betalar du med pris eller tid. I den här triangeln hittar du ganska mycket sanningar om vad det är som händer. Sen tror jag att omvärldsförändringar eller upptäckandet av behov, är sådant att man måste förändra utvecklingen medan den sker, annars så får man snapshot tänkandet.

### **Anna**

Ja där har man ett jätte stöd att RUP har ett så pass gott namn och nästan lite hippt att använda, som gör att det blir enklare att få med folk. Ofta är det ledningen som börjar med att vi ska profilera oss att vi ska använda oss av RUP, det är nästan på väg till som med ISO standarden Rational har verkligen lyckats. Egentligen är det lite lustigt för det är ju inget nytt. Speciellt för något år sedan i mindre utvecklingsföretag som utvecklade webbapplikationer där blev det ett sätt att kontext anpassa, helt klart. Så det är lätt att få stöd tycker jag.

### **Hur stor anser du att graden av acceptans är hos individer i projekten? Är det någon skillnad vilken roll det handlar om?**

#### **Petter**

Jag tycker att RUP ger stöd i alla roller där den fått minst genomslag är hos konstruktörer i tekniska projekt, många gånger vill de inte ändra på sig de är även så styrda i vilken miljö dom jobbar i t ex. om du jobbar i ett programspråk som inte är objektorienterat så blir det lite krystat med alla dessa aktiviteter, och du måste definiera om många av artefakterna, och dina guidelines, men jag tycker att det skiljer väldigt mycket det här med acceptansen från fall till fall.

#### **Jens**

Är man otränad på RUP eller att jobba i RUP projekt så kan det bli ett krig mellan rollerna att vissa roller är finare än andra. Att det blir lite tävlig i att samla flest roller. För du jobbar ju egentligen inte så annorlunda när du faktiskt gör någonting.

#### **Anna**

Ja det är alltid mest svårt med programmerarna, de är lite "kan själv". Samtidigt finns det här med UML och de flesta programmerarna inser att de kan bli ännu bättre om de vidgar sina vyer mot arkitektur. Det höjer ju individen värde om du kan säga att man kan RUP och UML. Så det är inget jätte problem egentligen. Acceptans handlar egentligen om att hitta ett projekt som är lagom stort lagom svårt för att komma i gång med.

## A.1.5 Reliability – reliabilitet

**Är processen uppbyggd så att process fel undviks eller blir funna innan de resulterar i produktfel? Vad anser du om processens tillförlitlighet?**

### **Petter**

Jag tror det handlar om att har du full frihet har du ingen kontroll, och har du full kontroll har du ingen frihet, och RUP ligger i sitt grundutförande mer åt, eftersom den är så stor och generell, om du inte gör anpassningar så har den definitivt väldigt lite av det här med process inherent quality tycker jag, den beskriver ju allt och så, men man kan ju hitta fel också man kan ju gå vilse i processen, men rätt implementerat i och med sina checkpoints, så är den väldigt bra på att leda dig rätt. RUP i sitt grundutförande är svår, och en av svårigheterna är storheten. I de fall när vi verkligen har lyckats är när vi haft en eller två visionärer i projektet som säger att jag har visioner lita på mig, koncentrera inte bara dig på att följa dina guidelines utan vi ska jobba utifrån våra förutsättningar och då funkar det väldigt bra. Det finns en missuppfattning att alla processer är lika, det räcker att gå intr. kursen och sedan läsa boken, det finns så mycket i RUP som är inbyggt svårt och bara för att RUP är så pass generell har man inte adresserat dessa svårigheter tillräckligt noga, så att du måste veta vad du gör, även som enskild utvecklare, åtminstone få hjälp med avsikten. Vet du inte varför du gör det kommer det att bli dåligt.

Processen kan egentligen bara styra innehållet - hur bra krav har jag skrivit genom att tillhandahålla verktyg det har jag exitkriterier för att verifiera. Just det här att man kliver över i en mer komplett beskrivning av krav via use-cases och andra hjälpmedel, det blir lättare för dig att skriva bra krav än bara skriva dessa krav textbaserat.

### **Jens**

RUP är ganska bra på att om man kombinerar flera verktyg att härleda i vilket use case eller i vilket sammanhang eller i vilken iteration där du inte är konsekvent mot vad du tidigare beskrivit, och det är ju inte samma sak som att det är fel utan det är fel gentemot vad du tidigare beskrivit, så på så sätt hjälper RUP dig att vara konsekvent, men har du tidigare börjat kalla apelsiner för bananer kommer inte RUP hjälpa dig och tala om det. Den hjälper till så att sammanhanget hänger ihop, den skapar coherence. Alla delar du gör allting du utvecklar är kopplat på det sätt du själv har beskrivit. Sedan om det är rätt eller fel, bra eller dåligt, det svarar inte RUP på. Men den svarar på om det stämmer med de förutsättningar du har med dig, så RUP hjälper dig att vara konsekvent i det du gör. Har man tänkt rätt från början så minskar ju felen, men det är inte processen i sig som gör det.

### **Anna**

Man kan säga att det här med milstolparna är ett sätt för att hitta fel så tidigt som möjligt. RUP är väldigt risk orienterad så redan i andra fasen ska man fokusera på det som påverkar arkitekturen, det som man bedömer vara mest riskfyllt. Det är det som man ska bygga då, där är processen bra för att den tvingar dig att ta tag i det svåraste. Så processen är bra på det sättet, följer man den är det ingen stor risk att man när man levererar beta realisen märker att det finns fel. Däremot finns det en mental spärr att börja med det svåraste först. Man lyckas nästan aldrig driva igenom detta tyvärr, om inte det visar sig att det svårast även är det viktigaste i projektet. Men är det så att man ska bygga en applikation som ska ex hantera material tillgång så har man ett användningsfall som säger att externa leverantörer ska kunna komma in och titta på status det är en typiskt arkitektur sak som kan påverka hela lösningen,

brandväggar osv.. Ett väldigt riskfyllt användningsfall men samtidigt en ganska liten funktion i systemet, frågan blir ska man ta denna först eller vänta med den. Processen ger stöd för att förhindra detta i alla fall.

Om det sker fel så beror det väldigt mycket på bristande kommunikation inom projektet eller mot kund. Eller om man slarvat med milstolparna om man inte har utbildat sin kund att förstå vad de här milstolparna innebär blir det problem. Och det är alltid mycket svårare i verkligheten än vad de är i utbildningen eftersom det handlar om pengar, så säger man ett pris på hela systemet och sedan skriver man ett avtal på det. Så börjar man med kravarbetet och så visar det sig att det inte alls stämmer ex att det är mycket mer och då förhandlar man om vid första milstolpen, där är det nästan alltid problem. Kommunikation är nog nyckelordet för att undvika problem, och att frysa kravbilder, så att kraven inte får förändras ända till slutet.

#### A.1.6 Robustness – robusthet

##### **Kan processen fortsätta trots oväntade problem? Hur ? Lösning av problemen?**

###### **Petter**

göra riktiga milestones assessment

Den stora vinsten är uppdelningen i arbetsflöden bar för att du har alla arbetsflöden öppna hela tiden att processen till sin natur är iterativ så har du skapat dig en mycket större stabilitet för oförutsedda händelser.

###### **Jens**

Den är väldigt hållbar eftersom man arbetar med ganska korta och målfokuserade insatser. Om man säger att man gett sig på den värsta risken när det gäller ett interface mellan två olika maskinmiljöer eller något liknande. Man har kriterier på svarstid och annat , där emellan, och man börjar få problem med just det use cases som hanterar detta får man lägga på fler iteration som knäcker detta , på så sätt så hanterar RUP oväntade problem eftersom den utgår ifrån en bild med att det här tror vi är en riskfylld sak, så till vida att den innerhåller okända faktorer, och därför ger man sig på den först, då får den ta den tid den tar, man lär sig ju mest inom det område där det är flest okända faktorer. På så sätt är den robust.

###### **Anna**

Det finns ingen direkt exeptions hanterare i själva processen, men vid milstolparna så mäter man ju hur långt man har kommit och man tittat på om det hat kommit upp några nya risker. Väldigt ofta i projektet stannar man upp och tittar, var är vi nu för någon stans, vart är vi på väg. Och där har man ju möjlighet att hitta på plan b redan till nästa iteration. För oftast gör man iterationer speciellt när man börjar närma sig slutet har man väldigt korta iterationer. För att kunna se att man rör sig framåt, så iterationerna är kanske på en vecka. Och då får man lättare att kasta om , men ingen ren katastrofplan.

##### **Vad anser du om processens hållbarhet?**

###### **Petter**

Mycket av det som finns i RUP växte fram under 90-talet det innebär ingen revolution, der är ju inget paradigm skifte det är bara en evolution , och mycket av det som finns i RUP är en dokumentering av det som vi i branschen visste, därför är den så hållbar som en process kan bli den kommer ju successivt fyllas på med mer och mer kunskap , och om det inte blir någon väsentlig förändring inom systemutveckling , så är RUP hållbar väldigt långt framåt . Det man

kan tendera till om man pratar om processens hållbarhet är ju att eftersom den är så artefakt tung , roll „väldigt mycket du måste ta fram i grundutförandet. Kan det ju bli så att den får lite elephantiasis, så skulle det komma någon förändring så ser jag två – en hårdvaru del så att du hanterar mer än bara programvaru system , och det är väl nödvändigt eftersom fler och fler system blir inbyggda, och den andra är en quick RUP , nu har vi en generell process som kan hantera projekt oavsett storlek , RUP för små projekt , snabb RUP med mindre verktyg en minivariant.

**Jens**

**Anna**

Den är så pass stor och generell så den står sig ju. Det som gäller är att man har förmåga att anpassa den efter sitt behov. Det krävs helt klart.

### A.1.7 Maintainability – varaktighet

**Anser du att det sker utveckling/förbättring av processen, från projekt till projekt?  
Hur?**

**Petter**

Det finns ju en utveckling ifrån Rational's sida. Och även i uml community, snart kommer nästa version av UML . där du har lagt på lite mer egenskaper som har att göra med tekniska system så där finns en kontinuerlig förbättring .Internt mäter vi ju inte körprocessen rakt av utan du kör på en anpassning av din företagsanpassning och att man pratar mycket om i processen att man ska återkoppla dina erfarenheter , så den är synnerligen lämplig att återföra förbättringar från projekt till projekt, att du ändrar ditt development case.

**Jens**

Rational är ganska aktiva med att lägga på de bitarna, de har bytt en del namn på diverse saker och händelser. Sedan är det en kunskapsfråga som är större, projektmedlemmarna är vana att arbeta på det här sättet. Och den biten som jag tror att alla som arbetar med RUP har är att skapa en förståelse hos kunden , vad det innebär att vara an beställare av ett projekt som utförs med RUP.

**Anna**

Ja man blir duktigare och duktigare. Man växer själv. Det finns ingen kunskapshanterare i processen. Processen utvecklas också det kommer uppgraderingar varje år från Rational. Så den fortsätter ju också att utvecklas.

**Hur sker dokumentation av processen? På ett lätt förståeligt sätt?**

**Petter**

Dokumentering av processen är åt det tekniska hållet, steg för steg, och ibland funkar det , men ibland kan det kännas lite tråkigt, i en mogen org. bättre göra en avsikts förklaring med en aktivitet och sedan strunta i hur man gör i detalj. Se bara till att alla förstår vad det är de ska göra , så kan du strunta i steg för steg. Det är mycket dokumentation i RUP , man rekommenderar att man ska ta fram mera- göra dina egna anpassningar och guidelines,

**Jens**

Om man kör RUP Online så har man ju verktyg som gör plats och detta är ett ganska effektivt sätt, vi har inte provat något annat. Och i den då skala ner och beskriva den process som gäller för projektet.

### **Anna**

På alla plan. Man kan se systemet som en databas full med information och UML notationen som ger en förmåga att se informationen från väldigt många olika håll, så med olika vyer kan man titta på systemet som är väldigt bra sätt att kommunicera med kunder och användare, programmerare osv.. Det är samma information men man kan titta på den på olika sätt. Så det är den stora styrkan, men det är ju inte bara RUPs förtjänst.

### **Vem är det som är ansvarig för dokumentationen?**

#### **Petter**

Det är olika ibland på projektet självt, just nu vi har en processingenjör som sköter dokumentationen, och i förra projektet var det metodavdelningen. Egentligen vill man att det ska vara en linje funktion utanför projektet, som sköter dokumentationen för att du vill lägga det på något som är mer persistent än ett projekt. Egentligen vill du hantera förändringar i processen på två sätt det ena är att du gör exception request för ett enskilt projekt, - vi tänker inte följa vår företags anpassning, just i det här fallet passar den inte. I Kvalitets aspekt måste du tala om att du gör den annorlunda , men i grund och botten tycker vi att grundprocessen är mycket bättre - så att du ändrar bara för projektet, idet andra fallet så gör du en chance request till den som är ansvarig för att underhålla processen och ändrar i företagsanpassningen dvs. vi höjer vårt undantag till regel i stället. Hur man hanterar det organisatoriskt egentligen är det du behöver är en som är ansvarig för företagets anpassning och en som är ansvarig för projektets anpassning.

#### **Jens**

Processingenjören, den ska sedan revideras när man går ur en iteration

#### **Anna**

Alla roller har sitt ansvar och det är definierat vem som har ansvar för vad, så systemanalysis har ansvar för vissa utpekade artefakter och arkitekten för något annat.

Arbetet med dokumentationen sker kontinuerligt, man dokumenterar inte vid sidan av utan man dokumentera för att produkten ska växa fram. Tack vare att man kan generera kod och så där, och sedan kan man utifrån kod även generera dokumentation. Så det är väldigt tätt knutet där, och sedan har man de där milstolparna som säger att då ska det finnas.

### **Finns det något annat sätt än att dokumentera processen för att bevara den? Vad? Hur?**

#### **Petter**

Ja du kan sätta en praxis, faktiskt , och det handlar om kunskapen, vi pratar ju om en hantverksskicklighet så att säga – hur många use case ska jag ha- antingen skriver du en regel eller så gör en tio femton hur hittar jag rätt på det här..bygger du in detta i din organisation – ja då behöver du ju inte dokumentera det kräver ju att du har en kontinuerlig kunskaps , bra kunskapsorganisation som hanterar detta.

#### **Jens**

Mycket av kunskapen finns nog i huvudet hos projektledaren och processingenjören.

**Anna**

Nej man måste dokumentera löpande.

### A.1.8 Rapidity – snabbhet

**Vad anser du om processens möjlighet att utveckla ett system från en specifikation?**

**Petter**

Den är god, det är väl till och med så att du skulle kunna utveckla processen från ett business dvs. en marknad från ett kundkrav, utan en specifikation. Snabbheten processen styr väl, det som hjälper dig mest att utveckla från ett specifikation och få ett resultat som är riktigt är ju det här att man när sant iterativ i processen, så har du förstått det iterativa elementet i RUP så är det en väldigt bra process. En sak som man har gjort bra i RUP är att man trycker på att alltid ska gå igenom alla arbetsflöden i en iteration, även om det är en exception. Även i de tidiga faserna säger man att man ska gå ner i implementation, och följer man det, och då pratar jag om avsikten att gå ner på sk kodflödet det är det som stödjer dig att leverera rätt saker i slutändan. Jag tycker att just den indelningen i workflows, faser och iterationer som är en av de absolut stora fördelarna med RUP.

**Jens**

Om man ser det som vi på Cap gör är att vi sätter ett datum när vi ska leverera någonting. Jag har inget bra svar om det går snabbare, men förmodligen gör det eftersom man är mer konsekvent.

**Anna**

Jättebra om man vet vad man gör. Men det beror helt på vilken erfarenhet man har ifrån början. För är man erfaren och kör krav, design, kod så fungerar det väldigt bra. Men ska man under resans gång lära sig processen, lära sig verktygen ja då är den inte så snabb.

**Hur ser Tidsplaneringen ut?**

**Petter**

Återigen är det iterativa tänkandet som hjälper dig att – man är lite otydlig för man säger att man ska bara göra en fasplan men sen ska du bara detaljplanera en iteration framåt. Jag har läst den instruktionen och håller med - fast jag har sett att den feltolkas – varje projekt jag har varit med så har man feltolkat så att man sätter upp ett slutmål och gör bara en iteration, du ser inte om det pekar mot slutmålet eller inte, man har inte lyckas förmedla den intensionen att du måste ju ha några way points också. Du kan inte bara sikta mot slutmålet.

**Jens**

För tidsplaneringen för ett helt projekt tror jag fortfarande lite hädiskt att det är summan av erfarenhet och fingret i luften syndromet. Har du erfarenhet ifrån projekt inom samma miljö, bransch osv.. då vet man bakgrundsfaktorerna ganska bra.

När man sätter upp iterationerna blir det en tydlighet i varje leveranspunkt, det blir små vattenfalls steg där varje steg är så pass kort så att du hinner ha överblick vad som händer från a till ö.

Genom att trycka ner det till dessa korta iterationerna, då jobbar man i en ganska hård timebox – vad är det som ska levereras kopplade till use case. Då vet du vad det är för artefakter som ska komma ut, vad du ska leverera under den här tidsperioden, och på så sätt blir det hårdare styrning av att du faktiskt levererar i tid. Den andra vinningen med RUP är



den delen de kallar integrationen som varje gång du levererar så integrerar du det med det föregående. Här hade vattenfallsmetoden ofta problem i slutet att de helt enkelt inte kunde integrera stora delar av systemet. Så i RUP ligger testandet utspritt över hela tiden.

#### **Anna**

Det ser ut som översikt bilden och sedan får man titta på hur komplext är det här problemet. Vad har vi för erfarenhet i projektet och utifrån det får man försöka göra någon vettig tidsplanering. Men jag försöker jobba så att man planerar, man estimerar den första milstolpen den kan man säga är hyffsad säker att man når dit, och de andra är uppskattningar och vid varje milstolpe så måste man faktiskt få lov att omförhandla till nästa milstolpe för man lär sig ju så mycket efter resans gång. Om problemområdet, så jag tror inte på fast pris per projekt.

#### **Vad anser du om processens effektivitet?**

#### **Petter**

#### **Jens**

Har man grupper av människor som är kunniga i ämnet så är det effektivare, men RUP har en ganska hög instegs tröskel. Samtliga projektdeltagare släpper frågorna från gamla beprövade arbetsätt. Och då finns det alltid risker med skugg dokumentation, ja vi gör så säger man på mötet ,men gör på ett annat sätt i praktiken, man levererar fortfarande samma resultat men kanske har kommit fram till det på ett sätt som inte var tänkt att man skulle jobba efter. Så för mig ställer RUP större krav på kunnigheten av processen hos samtliga deltagare än tidigare metoder. Tidigare metoder har inte lagt sig i så mycket hur du gör själv vid ditt eget skrivbord så att säga.

#### **Anna**

Den kan bli väldigt låg om man försöker ta hela kakan. Då finns det en risk att man bara processar och processar och det blir inget av det. Så man måste veta vad man gör och varför. Man får inte glömma att det man ska göra är att bygga någonting.

### **A.2 Hägerfors**

#### **Anser du att RUP gör det möjligt att välja tekniker och metoder för att utföra arbetet så att de överensstämmer med situationen, organisationen, personer och mål.?**

#### **Petter**

Nja, och det är väl lite hur man säljer RUP, RUP säljer man som architectuell scentric and use case driven, och i arkitekturen har man ett visst antal beskrivningsformer som är dina typer av UML diagram. För att visualisera arkitekturen. Tar du bort betydande delar av det får du svårt – det går men- jag har sett vad son händer om man skriver funktionella krav i stället för use cases och det blir svårt. Ja det finns möjligheter att välja andra tekniker och metoder, men jag tror inte man vill göra det.

#### **Jens**

Jag har lite svårt att svara generellt på det, men oavsett process eller metod man använder så är det ett krav att man ska kunna välja fritt. UML blir svårt att säga eftersom vad skulle man annars ha eftersom det är en etablerad standard. Men där efter bör man och ska kunna välja tycker jag.

### **Anna**

Ja det är precis det den gör, det är ett ramverk som kräver att man tänker till.

### **Anser du att teknikerna och metoderna används på ett kompetent sätt.? Hur?**

#### **Petter**

Jag tycker att det här med objektorientering i stort och i RUP projekt i synnerlighet är att det är för få som kan det. Det kan ha att göra med att man ofta hör inte av sig till oss om man inte har misslyckats en gång tidigare, vi brukar komma in när det är kris, så vi ser väl ofta bara de värsta fallen. Jag har sett vad som händer om man misslyckas, om du misslyckas ordentligt och det tyder på att det inte är självklart hur man gör rätt ännu.

#### **Jens**

Jag tror att faran med RUP är att det är så pass nytt, att de flesta projekt du ser inte än är andra eller tredje, man har i många fall inte gjort ett första försök än. Om man tittar på de som jobbar med RUP idag finns det väldigt få referenser på genomförda projekt hos samma kund eller verksamhet. Det gör nog att man har en liten nybörjarsituation, man tenderar att använda metoden i för stor omfattning, man försöker införa för mycket på samma gång, man kan inte bara vara isolerad till att lägga in RUP i projektet man måste även lägga in den i organisationens sätt att göra saker. Bedriver man inte förstudier och skapar use case eller kravbeskrivningar på den notationsteknik som är användbar i RUP, då har man till slut bara ett text dokument på 100 sidor som inte hjälper dig att starta.

### **Anna**

Det är väldigt olika i olika projekt så det är svårt att säga något generellt. Varje projekt är ju unikt. Det är ju väldigt situationsberoende, men RUP är bra på att man stannar och tittar upp vart man är på väg. Man eliminerar med RUP att man inte löser fel problem eftersom som man checkar av med användarna hela tiden.

### **Anser du att RUP fördelar arbetet på ett bra sätt med hänsyn till tillgängliga personer.? Hur?**

#### **Petter**

RUP är bra med det som gäller roller, man frikopplar rollerna i projektet från din organisation, så att du bryter lite ner barriärerna, att den bästa personen kan vara både analytiker och testare. Det tycker jag faktiskt, med varje roll följer ett syfte i projektet, du har en metodbeskrivning, och du har ett antal aktiviteter.

#### **Jens**

Det är en bedömningsfråga, det är där skickligheten hos processingenjören och projektledaren kommer in. Att bedöma vad behovet egentligen är och sedan fördela detta på rollerna. Rolltänkandet ger ett visst mått av frikoppling på hur man bemannar, man kan ha människor som går in och ur projektet på ett tydligare sätt än vad du har i andra metoder. Man utgår ifrån - vad är det för kompetens jag behöver och sedan letar man reda på den.

### **Anna**

Det tycker jag det är rollerna, och man kan se en roll som en hatt, och i ett projekt kan en person spela flera roller och då ser man utifrån översikt bilden var de olika rollerna är som mest aktiva. I stora projekt spelar många personer en roll, det är lätt att ska upp och skala ner,

och det är ju definierat vad varje roll har ansvar för. Det har varit bra i små projekt där man ofta har en informell ansvars fördelning då är det lätt att saker ramlar mellan stolarna. När man har en process som kräver att man definierar roller så kräver den ju också att vi definierar ansvar och detta kan var ganska skönt. Då är det skönt att luta sig mot en process.

### **Finns all behövlig kompetens tillgänglig ?**

#### **Petter**

Det finns många som har tillräcklig kunskap om RUP, men de väldigt duktiga dvs. visionärerna saknas fortfarande. Det är rätt besläende att inte änns de riktigt stora leverantörerna har tillräcklig med folk att skicka ut. Det är för få. Det finns för få bra kurser i RUP. Det finns även ett problem med att ,man måste vara certifierad av Rational för att kunna bli rekommenderad.

#### **Jens**

Ja hos oss tycker jag det, det som är kritiskt är ju de bitarna med processingenjören att kunna den rollen, den är så intimt förknippad med RUP i sig idag, där krävs det att man bygger upp erfarenhet och kunnighet för att klara av att kunna leverera RUP projekt till kunderna. Att var systemarkitekt per RUPs definition kan innebära att man får ändra på de metoder eller arbetssätt man har inom sin organisation, men det är mera att lära känna RUP och begripa vad det är som jag ska leverera, än att stödja ihop processen på ett sådant sätt att den just passar ett utvecklingsprojekt. Där ligger en väldigt tung roll

#### **Anna**

Det finns ju expertkonsulter från Rational att kalla in både vad som gäller process och vad som gäller verktyg. De är ganska dyra. Men som kompetensöverföring är det ganska bra att ta in någon som hjälper en att komma i gång. Men det finns även kompetens inom företagen eftersom det är lite hippt, eller så rekryterar man folk.

### **A.3 Checkland&Scholes**

#### *A.3.1 Efficiency*

### **Resulterar processen i den önskade produkten?**

#### **Petter**

Igen generell process hjälper dig att ta fram den önskade produkten, men RUP som iterativ process med modellstödet som finns och just det med att du har erfarenhets återkopplingen gör dig bättre på att ta fram rätt produkt. Det blir mycket lättare att göra produkten med en bra process och RUP är en bra process för att succesivt höja kvaliteten i en produkt.

#### Process/produkt relation

#### **Jens**

Den levererar en produkt med de kraven som man var överens om. Det hjälper den till att göra. Sedan om det är rätt eller fel, det tycker jag ligger utanför. Det är en av farorna med Rationals eget sätt att argumentera för den, man levererar rätt sak i någon vid bemärkelse, och det enda man gör egentligen är att man sträcker sig att man utvecklat något på ett bra och konsekvent sätt.

### **Anna**

Det är helt beroende på hur man lyckas använda processen. Det finns inget i processen som tyder på att det inte ska bli något annat en högkvalitativ produkt. Men det är helt beroende på situationen , vilken teknik man använder och vilken kompetens man har i projektet och risker osv..

Jag tycker att RUP som recept ger väldigt bra förutsättningar, men det är ju ingen garanti för att lyckas.

### *A.3.2 Efficiency*

#### **Används minimala resurser? Finns det andra sätt, som är billigare, t ex andra tekniker?**

### **Petter**

Det skulle bli billigare - har du de fundamentala delarna i RUP och visste du vilka du behövde och bara hade dessa. Så skulle det var billigare – men frågan är om – det är det med anpassningarna hängande över dig . RUP är väldigt dyr...

### **Jens**

Det finns en ganska stark koppling till storlek, och det finns en koppling till om det handlar om utveckling eller om det handlar om nyutveckling av något som redan finns. Är det utvecklat med RUP processen så finns det en viss typ av dokumentation som troligen gör att det går väldigt snabbt och smidigt att göra en vidareutveckling. Är projektet litet kan man ju säga - varför inte arbeta med XP? Är det utvecklar med gamla tekniker så passar kanske äldre metoder bättre osv.. RUP passar ju inte överallt.

### **Anna**

Ja det finns andra som är billigare, RUP är ganska resurskrävande. Jag har arbetat med undervisning i implementation och varför man ska välja RUP. Man kan välja att antingen göra ett projekt enligt en systemutvecklingsmetod eller att använda sig av ett hack,(slänger ihop en lösning)det skiljer ganska mycket både i tid och pengar mellan dessa olika tillvägagångssätt, men det är mycket större chans att lyckas med en metod. Ett hack ser billigt ut till en början men sedan mot slutet så uppkommer det ofta stora kostnader som man inte räknat med. Ex förändrade krav som kräver en annan arkitektur, man vet inte ens om man landar i en lösning för man vet inte om man bygger rätt sak. Man vet inte heller att man bygger saker rätt. Så det är inte säkert att det blir dyrare med RUP, för det är större chans att man går i mål.

### *A.3.3 Effectivness*

#### **Leder metoden till uppnående av långsiktiga mål, t ex ökad marknadsandel eller god arbetsmiljö?**

### **Petter**

En bra process som RUP ändå är, gör ditt arbete lättare och du får en större förståelse varför du finns i organisationen och du kan koncentrera dig på det som är svårt. Någon löser det här strukturerings och planerings problemet åt dig och du kan bygga system. Jag tycker att det ger en bättre arbetssituation.

### **Jens**

Jag tror att beställaren upplever så länge han inte är så insatt eller erfaren att RUP projekt är ganska obehagliga och ostyrbara eftersom man inte ser så mycket av de traditionella styrparametrarna. Jag tror att det stundtals är ganska hög belastning på de roller och personer

som sätter upp projekt för att få det att hänga ihop när man börjar, när det sedan rullar så erbjuder RUP en gemensam, och man är överens om begreppsapparat runt och i projektet. Vad är det vi pratar om och vad är det vi jobbar med. Och om man i någon vidare begrepp pratar om arbetsmiljön så blir den bättre.

#### **Anna**

Det ger förutsättningar för att uppnå det i alla fall.

#### *A.3.4 Ethicality*

#### **Hur anser du att processen uppfattas?**

#### **Petter**

Stor och komplex och i och med att man bara har fokuserat på programvara så i vår domän uppfattas den med att hänga ihop med IT system. Presentationen tycker jag gör – det har lite med att göra hur den marknadsför som ”everything under the sun” så uppfattas den lite av elephantiasis av många, sen vet jag bland kollegor som kan RUP, så uppfattas den som bra, det är lite som man säger om Microsofts produkter, jag gillar inte det de står för men , men jag använder den ändå. För det är der bästa alternativet, och alla andra använder det också.

#### **Jens**

Svår att lära sig, hög ingångs tröskel. Skapar osäkerhet än så länge eftersom det är ont om referenser hos samma kund. Jag tror att det har mycket med att göra man inte har samma kontroll parametrar som man hade tidigare inbyggt.

#### **Anna**

Den uppfattas av många som den bästa på marknaden. Både vad gäller process och verktygsstöd. Men samtidigt finns uppfattningen att det är dyrt och det är krångligt. Det finns andra alternativ men inte av den digniteten, men sen kan man ju fråga sig om behöver en process av den digniteten. Man kan säga att om alla i teamet har lika erfarenhet av problemområdet och har arbetat tillsammans och så där , då behöver man ju ingen process. Men man har en process för att det är aldrig så att alla är lika erfarna. Man har en process för att hjälpa en framåt. Man kan ju alltid snickra ihop sin egen process i projektet men det tar ju också tid och pengar. Så därför har den säkert blivit så populär för att man kan köpa den. Det finns företag som har en egen metod men då krävs det att det finns en metodavdelning som ständigt arbetar för att förbättra metoden, så genom att använda sig av RUP kan man säga att man outsourcar sin metodavdelning

#### *A.3.5 Elegance*

#### **Leder användandet av metoden till att verksamheten fungerar bättre och smidigare? Är processen utformad på ett tilltalande sätt?**

#### **Petter**

Det beror på - fortfarande träffar vi på tillräckligt många som inte kan ta sig in i processen den har på något sätt för många vägar in , och man har inte en vana av att göra adaptationer där du hittar rätt på en gång, sen det här med att de har bilder till allt är väldigt bra. Just det här när man tittar på flöden och roller artefakter ,och kopplar roller artefakter och aktiviteter till varan kors och tvärs det gillar jag, men jag tycker att man fortfarande har en viss barn sjukdom med det här med webb gränssnittet – men det är nog mera det här med det konceptuella webbhantering - hur lägger man upp en process på webben– och alla hittar rätt?

**Jens**

Man har paketerat RUP som en process som tål allt ifrån att 4 personer jobbar med något i 2 månader, till att 50 personer jobbar med något hur länge som helst. Det finns för mycket, det är som vilken traditionell metod som helst att du får fyra pärmor i handen, och då är det ingen som orkar arbeta efter den. RUP har lite samma syndrom, även om det ligger i maskinen så är det en fruktansvärd mängd information. Det är användaren som måste lära sig att skala. För att kunna detta krävs kunskaper och erfarenhet.

**Anna**

Ja det tycker jag, men det kräver att man använder den med förnuft. Den kan vara väldigt omständlig om man tar hela paketet. När man väl har bestämt sig för att använda sig av RUP hur ska vi använda RUP i detta projekt, man tar inte hela kittet och bara trycker på ett projekt. Det finns en bank där man kan plocka saker ifrån, men duktiga människor är fortfarande det viktigaste.