



EKONOMIHÖGSKOLAN
Lunds universitet

Institutionen för informatik

Rational Unified Process

I vilken omfattning har metoden RUP stött användarmedverkan tidigare, och metodens utveckling under tiden (Agila UP)

Kandidatuppsats, 15 poäng, inom systemvetenskapliga programmet

Framlagd: 2014-06-03

Författare: Sara Baker

Handledare: Anders Svensson,

Examinatorer: Magnus Wärja
Mirella Muhic

LUND UNIVERSITY
Informatics

Rational Unified Process

I vilken omfattning har metoden RUP stött användarmedverkan tidigare, och metodens utveckling under tiden (Agila UP)

© Sara Baker

Omfång: 78 sidor

Framlagd:

Handledare: Anders Svensson

Examinator:

Resumé

Uppsatsen har för avsikt att komma fram till i vilken omfattning utvecklingsmetoden RUP stödjer användarmedverkan. Genom att beskriva begreppen RUP, användarcentrerad systemdesign och användarmedverkan med hjälp av litteratur har jag kunnat sammanställa ett par frågor i avsikt att utföra en intervju med tre systemvetare. Resultatet från det empiriska materialet visade att systemvetarna upplever att RUP inte ger bra stöd för användarmedverkan. Utvecklarna upplever att användarna inte involveras tillräckligt främst i konstruktionsfasen. Främsta skälet till detta är att metoden inte ger möjlighet till att användarna ska involveras i projektet samt att användarna inte alltid har möjlighet att avsätta tillräckligt med tid för att kunna medverka tillräckligt i utvecklingen.

Abstract

The paper intends to arrive at the extent to which development methodology RUP supports user cooperation. By describing the concepts of RUP, user-centered system and user cooperation with the help of literature, I have been able to compile a few questions in order to conduct an interview with three computer scientists. The results from the empirical material showed that computer scientists feel that RUP does not provide good support for user cooperation. The developers feel that the users are not involved enough foremost in the design phase. The main reason for this is that the method does not allow for users to be involved in the project, and that users are not always able to dedicate enough time to participate enough in development.

Nyckelord:

Rational Unified Process, användarmedverkan, användarcentraliserad systemdesign, användare, användbarhet, och AUP.

Key words:

Rational Unified Process, user cooperation, users centralized system design, user, usability, and AUP.

Förord

Jag har fått hjälp från en rad olika personer som har haft lämpliga kunskaper inom mitt undersökningsområde. Jag vill rikta ett stort tack till de tre systemutvecklarna som har varit verksamma inom de tre företag jag har haft kontakt med. Utan deras hjälp hade jag inte kunnat genomföra den här uppsatsen. Jag vill även tacka Anders Svensson som varit min handledare.

Ett stort tack!

Sara Baker

Innehållsförteckning

1 Inledning	7
1.1 Bakgrund	7
1.2 Problemformulering	8
1.3 Syfte	9
1.4 Avgränsning	9
2 Metod	10
2.1 Tillvägagångssätt	10
2.1.1 Deskriptiv/induktiv metod	11
2.1.2 Litteraturinsamling	12
2.1.3 Litteraturkritik	12
2.2 Empiri	13
2.2.1 Val av undersökningsmetod	13
2.2.2 Val av respondenter	14
2.2.3 Genomförande av intervju	15
2.3 Analys	16
2.4 Reliabilitet och Validitet	17
3 Litteraturförankring	18
3.1 Användare	18
3.2 Användarmedverkan	19
3.3 Användbarhet	19
3.4 Användarcentrerad systemdesign	20
3.4.1 Iterativ process	23
3.5 Rational Unified Process	23
3.5.1 Best Practice	24
3.5.2 Faser	26
3.5.2.1 Förberedelsefasen (Inception)	27
3.5.2.2 Etableringsfasen (Elaboration)	27
3.5.2.3 Konstruktionsfasen (Construction)	28
3.5.2.4 Överlämningsfasen (Transition)	28
3.5.3 Arbetsflöden	28
3.5.3.1 Verksamhetsmodellering	29
3.5.3.2 Kravanalys	29
3.5.3.3 Analys & Design	30
3.5.3.4 Implementation	30
3.5.3.5 Test	30
3.5.3.6 Driftsättning	30
3.5.3.7 Projektledning	30
3.5.3.8 Konfigurations och ändringshantering	31
3.5.3.9 Utvecklingsmiljö	31
3.5.4 Användningsfall	31
3.5.5 RUP är mer Agilt idag	32
3.5.6 Tillämpning av Agila Unified Process AgileUP	
3.5.7 Arbetsflöden	34
3.5.8 RUP 2007 till 2014 – Vad har hänt?	36
4 Teorianalys	38
4.1 Förberedelsefasen	38
4.2 Etableringsfasen	39
4.3 Konstruktionsfasen	40

4.4 Överlämningsfasen.....	41
4.5 Sammanfattning av kapitlet.....	41
5 Empiri.....	43
5.1 Undersökning	43
5.2 Beskrivning av Företag 1	43
5.2.1 RUP användning	44
5.2.2 Respondentens syn på användare	44
5.2.3 Användarmedverkan i RUPs faser	44
5.2.4 RUP och användarmedverkan	45
5.2.5 Respondentens kritik mot RUP	46
5.3 Beskrivning av Företag 2	46
5.3.1 RUP användning	46
5.3.2 Respondentens syn på användare	47
5.3.3 Användarmedverkan i RUPs faser	47
5.3.4 RUP och användarmedverkan	48
5.3.5 Respondentens kritik mot RUP	49
6 Empiri.....	50
6.1 Undersökning	50
6.2 Beskrivning av Företag 3	50
6.1.1 RUP användning	51
6.1.2 Respondentens syn på användare	51
6.1.3 Användarmedverkan i RUPs faser	52
6.1.4 RUP och användarmedverkan	53
6.1.5 Respondentens kritik mot RUP	53
7 Analys och diskussion	54
7.1 RUP användning	54
7.2 Användarmedverkan i RUP:s faser	55
7.3 RUP och användarmedverkan	56
7.4 Respondenternas kritik mot RUP	57
8 Slutsats.....	59
8.1 Slut Ord	63
Bilaga 1	64
Bilaga 2	65
Referenser	76

1 Inledning

I detta inledande avsnitt har jag för avsikt att först presentera bakgrunden till uppsatsen. Jag diskuterar även problemformulering, syftet med uppsatsen, vilka avgränsningar jag har gjort samt vilka målgrupper uppsatsen riktar sig till.

1.1 Bakgrund

RUP utvecklades av världens mest kända förespråkare för objektorientering: Grady Booch, Ivar Jacobson, och James Rumbaugh. Rational var deras första projekt, när de började arbeta tillsammans tanken var att förena sina respektive grafiska notationer för objektorienterad systemutveckling till en enda, senare kallade Unified Modeling Language, UML. Syftet var att utforma en gemensam metod för systemutveckling (Lunell, 2003).

I senaste årtiondena så har dator användning ökat mycket både privat och yrkesmässigt enligt Marklund (2001). År 1997 var cirka 90 procent av kontorsanställda arbetade med datorer, i jämförelse med år 1989 då det var endast 60 procent. Denna ökning fortsätter fram tills idag och har orsakat att stora krav ställs på systemens design (Löwgren 1993).

Enligt Gulliksen & Göransson (2002) bör man engagera en rätt användare och vid rätt tillfälle genom aktiv användarmedverkan i systemutvecklingen om man vill lyckas med att utveckla ett användbart system.

Enligt Ottersten (2002) bör en produkt vara enkel och behaglig att använda, annars förlorar man användarna och eventuella kunder, många av utvecklingsmetoder fokuserar inte tillräckligt mycket på produktens användning, därför saknas alltid ansvar och kontroll för att få den högsta säkerheten av användbarhet. Löwgren (2004) anser med ett användbart system ska inte användarna behöva känna sig tvungna att använda det, och att systemet ska vara designade så att man kan använda utan tvång. Han menar också att ett system ska genererar en god attityd i människor genom att vara flexibelt och enkelt att lära sig.

Idag finns det många och olika mängder med system som inte funkar som det ska och till och med inte användbara (Ottersten, 2002).

Och det anser Gulliksen och Göransson (2002) också att många företag satsar med stora summor pengar på olika system som i slutändan inte fungerar, detta beror på att många företag tillämpar inte användarcentrerad systemdesign utan de fokuserar sig mer på systemlösningar istället, det betyder att användarna involveras inte tillräckligt mycket i systemutvecklingen, och det som görs i de flesta traditionella systemutvecklingsmetoder.

Många år senare var Rational Unified Process (RUP) en av de mest användbara utvecklingsmetoder som är byggdes på objekt-orienterad och ingenjörsmässigt angreppssätt. Men på senare år har många företag börjat använda Agila metoder eller Agila vid sidan om RUP för utvecklings projekt. Idag finns det många olika Agila metoder som fokuserar sig mycket på människan och teamet, metoder har utvecklats under en längre tid och bygger på många års erfarenhet, kommunikation och samarbete mellan medlemmarna. (Stone et al, 2005 Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules).

1.2 Problemformulering

Med ovan nämnda bakgrund anser jag att det är intressant att undersöka hur metoden RUP har stött användarmedverkan tidigare, och hur mycket stödjer RUP användarmedverkan idag. Detta avser jag att göra med en fallstudie på tre företag som använder sig av RUP/ AUP mina huvudfrågor är följande:

- I vilken omfattning har metoden RUP stött användarmedverkan tidigare jämfört med dagsläge?
- Hur såg RUP på användarmedverkan 2007?
- Hur ser RUP ut idag? Alltså metodens utvecklig under tiden (Agila UP)

1.3 Syfte

Syftet med uppsatsen är att komma fram till i vilken omfattning metoden RUP har stött användarmedverkan tidigare, och hur mycket stödjer RUP användarmedverkan idag. Det vill säga i vilken omfattning användarna involveras i ett projekt där RUP används. Med användare menar jag de som kommer att använda systemet för att genomföra uppgifter i arbetet eller i andra sammanhang det vill säga slutanvändarna, inte de som beställer systemet. Syftet med detta arbete är att undersöka dagsläget och dess rådande trender gällande hur upplevs användarmedverkan i RUP?

1.4 Avgränsning

På grund av ämnets storlek och för att möjliggöra fokusering på det syftet jag har presenterat kommer jag i detta arbete att göra följande avgränsningar. Jag avser inte behandla eller jämföra UML modell, tillämpning av systemarkitektur, RUPs centrala modelleringselement samt tekniska detaljer kring implementation i RUP. Eftersom detta kan leda till en ny bok om RUP avser jag inte att omfattande beskriva RUP utan endast de delar som är nödvändiga för att ge läsaren introduktion till vad RUP är och står för samt i vilken omfattning metoden har stött användarmedverkan och jämföra det med dagsläge. Jag avser vidare att begränsa arbetet till att undersöka två företag som använder RUP i sitt utvecklingsprojekt, och för att kunna komplettera, och följa upp min undersökning (2014)över tid, gjorde jag en ytterligare undersökning på ett företag som använder båda RUP och Agila metoder i sitt utvecklingsprojekt, då är det mycket tidskrävande att undersöka fler företag. Jag avser inte behandla alla delar i Agila metoder utan endast de delar som är nödvändiga för att ge läsaren introduktion till vad AUP är och står för samt i vilken omfattning metoden har stött användarmedverkan och jämföra det med RUP.

2 Metod

I detta kapitel avser jag att ge en förklaring till hur jag har gått tillväga i mitt arbete. Första avsnittet behandlar mitt tillvägagångssätt och vilken metod jag använt mig av. Under följande avsnitt presenterar jag litteraturinsamlingen. I de två nästkommande avsnitten har jag tagit upp varför jag har valt intervju som undersökningsmetod samt hur jag gick tillväga inför valet av respondenter. Kapitlet avslutar med en beskrivning av hur analysen kommer att utföras samt vad termerna validitet och reliabilitet innebär.

2.1 Tillvägagångssätt

Som inledning till mitt metodval väljer jag att göra en del liknelser med hur Bryman (2011) urskiljer olika strategier och ansatser. Uppsatsens teoretiska grund består av litteratur i form av böcker och elektroniska källor. Jag använder således befintligt material för att bygga den teoretiska grunden om RUP, användarcentrerad systemdesign och användarmedverkan. Med detta menas att jag till en början med hjälp av befintlig litteratur förklarar vad RUP, användarcentrerad systemdesign och användarmedverkan är. I min undersökning har jag utfört en litteraturgranskning och en intervjustudie som gjordes på två olika tidpunkter. Uppsatsens material har jag samlat mellan 2007 och 2014. I denna studie vill jag jämföra en undersökning som gjordes 2007 (angående användarmedverkan i RUP samt i vilken omfattning stödjer RUP användarmedverkan) med en ny undersökning från 2014, som granskar samma problem.

Intervjustudien består av noggrant utformad intervju med för ämnet väsentliga frågor (bilaga 1,2). De två första intervjun har utförts år 2007 på två olika systemvetaren i två olika företag som hade använt RUP i sitt utvecklingsprojekt, den tredje intervjun har utförts 2014 med en systemvetare som använder RUP också i sitt utvecklingsprojekt.

Undersökningsstrategin jag använder mig av i denna studie hoppas jag därför leda fram till i vilken omfattning metoden tidigare (2007) gav stöd för användarmedverkan i systemutveckling, hur detta stöd uppfattas nu (2014), och uppsatsen försöker också behandla

differenser i teoretiska och praktiska tillvägagångssätt när det gäller användning, och tidens gång.

Enligt Bryman (2011) är min undersökningsinsats av semistrukturerad karaktär. Jag är således intresserat att ställa RUP mot användarcentrerad systemdesign för att på så sätt kunna se hur mycket RUP tar hänsyn till användarmedverkan i ett utvecklingsprojekt. För att kunna få bättre bild av vad utvecklare har för generella tankar och synpunkter kring RUP och användarmedverkan och för att uppnå viss sannolikhet för att resultatet skall bli trovärdigt, ökade intresset för i vilken omfattning metoden RUP stödjer användarmedverkan. Detta är det främsta syftet till att jag väljer en undersökningsinsats av semistrukturerad karaktär. Jag har alltså utfört en intervjustudie vilken empirien utifrån sedan kommer att behandlas. I studien har jag även analyserat RUP utifrån användarcentrerad systemdesign. Detta för att ge min syn på hur användarmedverkan ser ut i RUP jämfört med innan. För att uppnå resultat och skapa en helhet ämnar jag alltså att till en början förklara vad RUP, användarmedverkan och användarcentrerad systemdesign är.

2.1.1 Deskriptiv/induktiv metod

Med att tillämpa en induktiv metod menas att man tar fram empirisk information, det vill säga sådant material som förr inte har redovisats. Det går ut på att en ny teori ska kunna ta form och därigenom bidra med ny kunskap. I kontrast till den induktiva metoden kommer den deduktiva metoden från redan existerande teorier med ett ändamål att pröva och utreda denna information ytterligare (Bryman, 2002).

Både den induktiva och deduktiva metoden kan sedan kombineras med att antingen vara normativ eller deskriptiv. Normativ metod handlar om hur någonting bör vara, deskriptiv metod däremot handlar om hur något är (Bryman, 2002). I denna uppsats används en deskriptiv metod för att uppnå mitt syfte. Jag beskriver således i vilken omfattning metoden RUP stödjer användarmedverkan i tre företag vid två olika tillfällen, därigenom har jag satt min utgångspunkt i empirin vilket får till följd att uppsatsen ligger närmare en deskriptiv/induktiv form än en deskriptiv/deduktiv. En normativ ansats hade för mig varit omöjligt eftersom denna innebär att jag i så fall skulle behöva utreda hur det bör vara, vilket inte alls följer linjen för uppsatsen (Bryman, 2002).

2.1.2 Litteraturinsamling

Med hjälp av de avgränsningar jag har gjort har jag kunnat välja litteratur som främst behandlar användare, användarmedverkan, användbarhet, användarcentrerad systemdesign och RUP. Med hjälp av denna litteratur kan jag på en teoretisk plan förklara vad begreppet RUP är, samt ge en övergripande förklaring till vad användare, användarmedverkan, användbarhet och användarcentrerad systemdesign innebär.

Litteraturen består av elektroniska källor, artiklar, och böcker som är samlat mellan 2007 och 2014. Den största beståndsdelen är böcker, med detta menar jag till antal. Jag har i erkända källor såsom andra uppsatser inom ämnet samt böcker om RUP, användare, användarmedverkan, användbarhet, användarcentrerad systemdesign, hittat dels författare som jag har valt att direkt ta fasta på men även genom dessa författares egna källor hittat andra erkända författare. Två av de böcker som jag har tagit fasta på är Gulliksen och Göransson (2002) ”Användarcentrerad systemdesign” och Lunell (2003) ”Fyra rundor med RUP”. Jag anser att Gulliksen och Göransson (2002) beskriver användarmedverkan och användarcentrerad systemdesign på ett pedagogiskt sätt samt att Lunell (2003) beskriver omfattande om RUPs faser och arbetsflöden. Genom att använda välrenommerade källor kan kravet på trovärdighet, meningsfullhet och representativitet uppfyllas (Bryman, 2002).

Alla elektroniska källor och böcker som använts en gång i tiden för denna uppsats har nog undersökts, granskats och uppdaterats, och de delar jag ansåg var förklarande och bidragande för den ”helhet” jag i denna undersökning försöker nå har valts ut för vidare studier. Jag gick igenom gamla materialen och jämfört med de befintliga, syftet var att hämta fakta och material från existerande litteratur angående RUP, användbarhet, användarmedverkan och användarcentrerade systemdesign.

2.1.3 Litteraturkritik

Jag är medveten om att den litteratur jag har valt är präglad av författarens bakgrund. Easterby-Smith et al. (2002) menar att den världsbild en individuell författare borde ses som en viktig faktor vilken framförallt kan påverka författarens val av undersökningsmetod, detta trots att författaren är erkänd. Jag anser att den litteratur jag har valt har en objektiv prägel, eftersom jag inte uppfattar den som extrem åt något håll. Detta är önskvärt för att minimera riskerna för att jag ska påverkas av författaren. Det är även önskvärt för att på ett neutralt sätt

kunna belysa negativa och positiva egenskaper gällande RUP och användarcentrerad systemdesign om det behövs. Att hitta litteratur som är helt opåverkad av författarens sinne eller av ovanstående anledningar ser jag är en omöjlig uppgift. Jag har dock med den vetenskapen Easterby-Smith et al. (2002) presenterat kunnat ta hänsyn till denna prägel genom att inte använda de delar av litteraturen där författaren omotiverat eller grundlöst trycker på sin ståndpunkt. Jag har istället valt de delar av litteraturen jag anser är förklarande, pedagogiska och bidragande till "helheten".

2.2 Empiri

En stor del av min uppsats bygger på empiriskt material som jag har samlat in och sedan analyserat. Jag har undersökt i vilken omfattning metoden RUP stödjer användarmedverkan under en utvecklingsprocess och i två olika tidpunkter. I en empirisk studie söker man information och data baserat på erfarenheter i den verkliga världen (Denscombe, 2000). Respondenterna jag har undersökt är tre systemutvecklare som jobbar på tre olika företag där metoden RUP används som utvecklingsmetod. Skälet till att jag enbart valde intervjua två systemutvecklare i fas ett (2007) och en i fas 2 (2014) är att det är svårt att komma i kontakt med systemutvecklare som använder sig av metoden RUP, för att systemutvecklare avstår att diskutera kring metoden då metoden är komplex.

2.2.1 Val av undersökningsmetod

Enligt Bryman (2002) och Bryman (2011) är intervjun sannolikt den mest använda metoden i kvalitativ forskning. Det beror på dess flexibilitet. Bryman (2002) delar intervjuer i två huvudsakliga kategorier, den ostrukturerade och den semistrukturerade. I båda fall är intervjuprocessen flexibel, speciellt i förhållande till kvalitativa intervjuer med strikta frågescheman. Tonvikten ligger i hur intervjupersonen uppfattar och förstår frågor och händelser.

Den undersökningsmetod jag har valt att använda mig av är kvalitativa semistrukturerade intervjuer. Enligt Bryman (2002) bygger en semistrukturerad intervju på att man utgår från några få specifika frågor för att vidare utveckla en diskussion med den intervjuade. Jag har utfört intervju för att på så vis, med hjälp av teori om RUP, användarcentrerad systemdesign och användarmedverkan, visa i vilken omfattning stödjer RUP användarmedverkan nu och

hur mycket var det innan. Att jag har valt kvalitativ semistrukturerad intervju beror på att jag ville få en bättre bild av vad den intervjuade hade för generella tankar och synpunkter kring användarmedverkan och RUP. Dessutom kunde jag via intervjuer skapa diskussioner kring området som skulle kunna ge mig mer information. Eftersom jag inte visste vad respondenterna skulle svara på vissa frågor ville jag ha möjlighet att ställa följdfrågor om respondenterna var inne på intressanta spår, vilket man inte kan göra exempelvis med enkäter.

2.2.2 Val av respondenter

I 2007 fick jag tag på två respondenter från två olika företag som passade min tänkta undersökningsprofil genom att använda de kontakter vänner och bekanta till mig hade på företag. År 2014 fick jag tag på en respondent från ett företag som jag själv har jobbat innan. Detta ses som styrande val och kallas enligt Bryman (2011) för bekvämlighetsurval. Jag var medveten om detta men beslutade att fortsätta med undersökningen eftersom det inte påverkar resultaten då jag dels inte känner respondenterna personligen, dels därför att mitt undersökningsobjekt är en systemutvecklings ansats, och inte systemutvecklare personlig.

Respondenterna kontaktades via telefon för bokning av intervju. De fick i samband med bokningen reda på att intervjun skulle vara mellan 30 minuter och en timme. Detta gjorde jag för att respondenterna har rätt till att bestämma över sin medverkan gällande samtyckeskravet (Bryman, 2002). Respondenterna fick även reda på att intervjun skulle handla om metoden RUP och användarmedverkan i synnerhet. En kort resumé av innehållet av intervjufrågorna gick igenom. Syftet med detta var att jag ville informera om att deras deltagande var frivilligt och de hade rätt till att avbryta om de ville.

Enligt Bryman (2011) bör även konfidentialitetskravet uppfyllas. Konfidentialitetskravet innebär att uppgifter om respondenterna skall behandlas med största möjliga konfidentialitet för att minska risker för att obehöriga skall kunna utnyttja informationen. Jag var medveten om detta, vilket jag förklarade för mina respondenter. Jag har inte för avsikt att presentera namnen på de företag jag använt mig av i min undersökning. Jag har namngett företagen som Företag 1, Företag 2, och Företag 3 samt använt respondenternas förnamn. På det sättet blir anonymiteten hög och detta anser jag kunna gynna reliabiliteten och validiteten i

undersökningen, något som även Bryman (2002) beskriver, man vågar säga vad man tycker och tänker på ett ärligare sätt då man är anonym.

2.2.3 Genomförande av intervju

Enligt Flowerdew & Martin (2005) bör intervjuer genomföras på de platser som respondenterna väljer för att de skall känna sig så avslappnade som möjligt. Av denna anledning lät jag intervjupersonerna bestämma platsen där intervjun skulle genomföras. För att det empiriska materialet skulle bli så korrekt som möjligt följde jag Brymans (2002) intervjuanvisningar och spelade därför in samtalen. Anledningen till att jag spelade in intervjuerna var för att jag på så sätt kunde fokusera mig på respondenterna samt att en inspelning ger större möjlighet till att upptäcka vad intervjupersonerna verkligen har sagt, vilket man inte alltid uppfattar när man antecknar. Jag gjorde en kort och tydligt förklaring för syftet av min undersökning innan jag börjar intervjun. Jag ville helst intervjua respondenter som är systemvetare och har erfarenhet av RUP eller haft omfattande roll inom metoden RUP. Intervjuerna höll på i 30- 40 minuter, och besöket varade i mer än en timme.

Intervjuerna transkriberades kort efter intervjutillfället, dessa skickades sedan till respondenterna för godkännande. Transkriptionerna finns att läsa som bilagor (bilaga 2). Att jag transkriberade intervjuerna kort efter samtalen var för att hålla kvar en hög kvalitet genom att jag då fortfarande hade intervjuerna i färskt minne. Att transkribera innebär att man skriver ut de inspelade samtalen ordagrant i löpande text. Innan jag började transkribera bestämde jag hur intervjuerna skulle skrivas ner. Jag har försökt undvika fel som kan bero på dålig inspelningskvalité eller att jag har hört fel genom att lyssna på inspelningen och ordagrant skriva ner vad som sägs. Sedan har jag ytterligare minst en gång lyssnat på inspelningen och samtidigt läst texten, och då det behövts stannat inspelningen och rättat eventuella felaktigheter i texten. I vissa stycken när den intervjuade pratar väldigt fort eller då något ljud i bakgrunden stör inspelningen, är det svårt att höra och då har jag lyssnat om och om igen tills jag har hört vad som sägs. När man översätter talspråk till skriftspråk, kan det ordagranna ibland låta förvirrat och uppfattas som osammanhängande. Därför har jag i vissa fall valt att redigera uttalanden och citat så att de blir läsbara och sammanhängande. Detta för att undvika vad som ibland kallas oetisk stigmatisering (Kvale, 1997). Intervjuerna har sammanställts i empirikapitlet.

2.3 Analys

I analysen är det viktigt att läsa och förstå källan noga, därför ordnade jag alla material jag fick från empirin för att kunna få en känsla för helheten, och att kunna göra jämförelse med teorin och försöka få tolkning till min undersöknings frågeställning. Enligt Svenning,(2003) är kvalitativa analysen som ett iterativt lopp där man går genom sitt material om igen hela tiden för att kunna komma fram till nya synpunkter och aspekter, därför när jag pratar om kvalitativa, menar jag att något är viktig och bra. Jag försökte använda detta i min analys genom att ha arbetet i tre olika faser Enligt Svenning (2003) är det öppen kodning, axiell kodning och selektiv kodning.

Öppen kodning innebär att man läser igenom materialet och för lite anteckningar som man anser är viktiga för studiens syfte. Det andra stadiet, axiell kodning, innebär att man försöker gå djupare in empirimaterialet för att kunna urskilja samband till kategorier som hör ihop i materialet. Därefter går man vidare till selektiv kodning vilket innebär att man selektivt försöker hitta detaljer och exempel som är kopplade till kategorierna som sedan kan plockas ut till slutsatserna. Med detta metoder i analysen har man stor möjlighet att lyckas hitta mönster och relationer som kan kopplas samman för att se om några slutsatser kan dras av det insamlade arbetet (Svenning, 2003).

Första steget var att jag sammanställa intervjumaterialet och läste de ett antal gånger samtidigt som jag gjorde en del anteckningar kring synen på användare, användandet av RUP, användarmedverkan i de olika faserna samt vad respondenterna tyckte om kritiken mot RUP. Nästa steg försökte jag gå djupare i de anteckningarna och se vad respondenterna svarade inom berörda områden, för att se om några samband kunde dras, jag började sammanställa vad de hade svarat. Här kunde jag se lite samband i svaren från respondenterna. I det sista stadiet satte jag ihop områdena som jag selektivt hittat i empirimaterialet, och på det sättet kunde se vilka slutsatser drog jag av de två olika tillfällena. Analysarbetet gjorde jag i iterationer och jämförelse ända fram tills jag kände mig nöjd och inte kunde plocka ut fler slutsatser.

2.4 Reliabilitet och Validitet

Reliabilitet och validitet används som ett forskningskriterium när man bedömer vetenskaplig forskning. Med validitet vill man se om den studie man har gjort verkligen mäter det man avsåg att mäta. Inom validitet kan man skilja mellan yttre och inre validitet. Yttre validitet innebär till vilken grad forskningsresultatet är användbart för andra situationer utanför detta forskningsexempel. Inre validitet innebär hur väl teorin och empirin är kopplade till varandra (Bryman, 2002). Med reliabilitet däremot vill man se hur väl studien är replikerbar. Replikering innebär att det är möjligt att genomföra en likadan studie och även uppnå samma resultat (Bryman, 2002).

Jag anser att min undersökning inte är så svår att replikera, eftersom jag har dokumenterat och redovisat hur jag har gått tillväga i varje steg, men det som ska diskuteras är om samma resultat skulle uppnås. Detta ställer jag mig frågande eftersom företagen förändras ständigt och nya metoder utvecklas. Samtidigt tror jag att kunskapsspridning om användarcentrerad systemdesign, användarmedverkan och användbarhet har ökat inom den sista tiden. Enligt Gulliksen och Göransson (2002) har forskning kring dessa ämnen ökat under de senaste decennierna för att minska de problem som uppstår i samband med utveckling och införandet av IT-stöd i arbetslivet.

För att uppnå en hög validitet bör metoder användas som undersöker det fenomen som reflekterar verkligheten (Bryman, 2002). I min undersökning genomfördes tre intervjuer. Intervjun är ämnad för denna typ av metodprovning vilket i sig hjälpte mig att fokusera på viktiga delar som tillsammans kunde utgöra ett helhetsperspektiv. På detta sätt anser jag att jag har undersökt det som jag hade avsikt att undersöka. Detta har bidragit till att jag hållit mig till undersökningens syfte och skapat en hög validitet kring studiens ändamål.

3 Litteraturförankring

Detta kapitel inleds med en beskrivning av vad användare, användarmedverkan och användbarhet innebär för de som använder RUP. Anledningen till att jag valde att förklara de här begreppen är för att få en så bra grund som möjligt samt få en förståelse om vad användarcentrerad systemdesign innebär. Detta för att sedan kunna ställa användarcentrerad systemdesign mot RUP och göra en jämförelse för att se skillnader de emellan. Kapitlet fortsätter med att jag förklarar vad RUP innebar och vad den innebär nu, och vilka granskningar finns idag när det gäller användarmedverkan i RUP, därför har jag utökat litteraturen med nya material från moderna böcker, artiklar och undersökningar som gjordes nyligen om RUP.

3.1 Användare

En systemutvecklare är inte den typisk användare av de system som hon eller han utvecklar. Om jag inte lyckas förstå vilka uppgifter användarna har eller hur uppgifterna utföras de, kan jag inte utveckla ett system eller en produkt som möter deras behov och förväntningar. Att förstå de tilltänkta användare, är en nyckel till hela utvecklings process (Gulliksen och Göransson 2002). Detta innebär att man tar reda på behoven från lämplig användare. När man pratar om användare så är det bra att veta vem man egentligen talar om. Är användaren, den som handlägger ett ärande eller är det kunden. Användarna är ingen homogen grupp, därför är det viktigt att skilja mellan domänexperter och verkliga användare. Domänexperter är väl insatta i verksamheten men inte representativa som slutanvändare. Domänexperter involveras kontinuerligt under hela utvecklingsprojektets gång. Slut användare däremot bör involveras för mer tillfälliga aktiviteter under analysen och primärt för utvärderingar av diverse designlösningar. Enligt Gulliksen och Göransson (2002) är användare de som kommer att interagera med systemet för att genomföra uppgifter i arbetet eller i andra sammanhang dvs. slut användarna, inte de som beställer systemet. Ottersten och Berndtsson (2002) skriver att användarna är de som har kunskap om tillämpningsområdet. När jag talar om lämplig användare utgår jag från Gulliksen och Göranssons (2002) begrepp, det vill säga att jag menar slut användaren. Användare i ett projekt kan etableras genom beställning av ett system eller med cheferna till personer som i slutändan är de som ska få använda systemet. Hur man ska

använda systemet samt hur man ska samverka med användarna beror väldigt mycket på vilka användarna är. Ibland medverkar användarna i ett projekt en längre tid vilket medför att de förvandlas från att vara en representativ, “typisk” användare till att bli en projektmedlem (Gulliksen & Göransson, 2002).

3.2 Användarmedverkan

Begreppet användarmedverkan betyder att användaren ska uppfattas som en fullständig medlem i utvecklingsgruppen under hela utvecklingsprojektet. Anledningen till att användarna bör medverka under hela projektet är att användarna är de enda som vet exakt hur slutprodukten ska vara och i vilken miljö och omgivning produkten ska verka i (Gullikson & Göransson, 2002). För att lyckas med att skapa bra produkter behöver man ha kunskap om hur användarna och utvecklarna ska bedriva utvecklingen. Enligt Andersen (1994) finns det tre olika strategier för hur experter och användare ska utnyttjas i utvecklingsarbetet. Dessa tre strategier är: expertdominerad systemutveckling (kallas även expertmodell), användarledd systemutveckling och samarbetssystemutveckling. Den expertdominerade systemutvecklingen innebär att utvecklarna utför arbetet helt utan användarna, användarledd systemutveckling innebär att användarna bedriver utvecklingen (Andersen, 1994). Den tredje strategin samarbetssystemutveckling, går ut på att utvecklarna och användarna samarbetar under hela utvecklingsprocessen. Denna strategi liknar Gullikson & Göranssons (2002) vid användarcentrerad systemdesign. Gulliksen och Göransson (2002) påpekar att det behövs projektlinjer för hur användarna ska delta i projektet. Projektlinjerna är av tre strategier, riktlinjer för användarmedverkan, riktlinjer för hantering av återkoppling från användarna samt riktlinjer för urval av användare. Riktlinjerna är ganska generella till sin natur och kan därför användas utanför tillämpningsområdet.

3.3 Användbarhet

Begreppet användbarhet betyder att ett system ska vara anpassat till de användare som ska använda systemet, det vill säga systemet ska vara användbart (Gulliksen & Göransson, 2002). Många organisationer använder luddiga och otydliga definitioner när de talar om användbarhet (Gulliksen & Göransson, 2002). Gulliksen och Göransson (2002) skriver att organisationer som använder sig av otydliga begrepp gör det för att då blir det svårare att ställa krav på utvecklingsarbetet.

Enligt Gulliksen och Göransson (2002) samt Andersen (1994) behöver även beställarens behov tillgodoses för att en produkt i slutändan ska vara användbar för alla användare. Detta anser de är viktigt därför att beställarens behov ligger på en högre företagsnivå där syftet med informationssystemet till exempel kan vara att göra så att företaget ökar och effektiviserar sin försäljning för att vinna marknadsandelar. Löwgren (1993) skriver att användbarhet även kan betecknas som ett resultat av anpassning, acceptans, effektivt och lärbarhet. Anpassning står för hur bra systemet stödjer användarnas behov, acceptans innebär hur användarna känner för systemet och effektivitet och lärbarhet står för hur effektivt användarna kan utföra sina uppgifter med hjälp av systemet. Dessa fyra beteckningar är utifrån användarnas perspektiv. Enligt Andersen (1994) användarna skall involveras för att klar göra sina behov, önskemål, testa och känna av systemet och för att klar göra de yttre egenskaperna, han menar att det finns ansvar som ligger på båda utvecklarna och användarna, utvecklarna skall involvera användarna för att förstå deras behov och önskemål och användarna ska bestämma vilka behov och önskemål vill de ha. Det vill säga utvecklarna bör använda användarna på ett korrekt sätt i utvecklingen för att kunna uppfylla kraven på de yttre egenskaperna. Även användarna har ett ansvar att hjälpa utvecklarna för att finna de yttre egenskaperna.

3.4 Användarcentrerad systemdesign

Att utveckla ett system genom att tillämpa användarcentrerad systemdesign är en attityd och filosofi som Gulliksen och Göransson (2002) förespråkar. För att lyckas utveckla en användbar produkt bör användarna medverka aktivt under hela utvecklingsprocessen, vilket är centralt inom användarcentrerad systemdesign.

Enligt Gulliksen och Göransson (2002) grundades användarcentrerad systemdesign 1986 då Norman och Draper släppte boken ”*User Centered System Design*”. Gulliksen och Göransson (2002) skriver även att användarcentrerad systemdesign har sin grund i människa data interaktion (MDI). Med MDI menas hur interaktionen mellan människan och datorer påverkas av varandra. MDI tar även upp hur stor nytta slutanvändarna kan ha av en produkt.

Författarna Löwgren och Stolterman (1998) skriver att utvecklarna inte har de kunskaper som användarna har om det arbete systemet ska stödja och inte heller om den miljö i vilken systemet ska verka i. För att kunna få denna kunskap anser författarna att det är viktigt att involvera användarna så mycket som möjligt under utvecklingen av ett projekt.

Definitionen av användarcentrerad systemdesign är enligt Gulliksen och Göransson (2002) att användarcentrerad systemdesign är en process som fokuserar på användare och användbarhet genom hela utvecklingsprocessen och genom hela livscykel. Vidare säger författarna att genom att följa användarcentrerad systemdesign det vill säga genom att låta användarna medverka under hela utvecklingsprocessen så kommer man att lyckas utveckla en produkt som är användbar. Användarcentrerad systemdesign kräver effektiv användarmedverkan. Med effektiv användarmedverkan menas att det ska vara rätt användare som deltar i projektet vid rätt tillfälle.

Gulliksen och Göransson (2002) grundar sin definition på tolv nyckel principer. Dessa nyckel principer befäster en korrekt och lämplig användarmedverkan i utvecklingsprojektet. Nyckelprinciperna är enligt författarna inte en modell för hur en utveckling ska gå till utan det är en process och ett tankesätt som utvecklarna kan ha till hjälp. Nedan ges en kort beskrivning av de tolv nyckelprinciperna:

Användarfokus - alla i projektet måste förstå verksamhetens mål, grunden i användarnas situation, deras uppgifter, varför och hur de utför sina uppgifter, etc. Det är viktigt att man prioriterar vad som är bra för användarna, framför vad som är tekniskt möjligt. Aktiviteter som att ta fram användarprofiler, kontextuella intervjuer och uppgiftsanalys måste vara en naturlig del av utvecklingsprocessen. En kontinuerlig fokus på användarna och deras arbetsuppgifter kan exempelvis åstadkommas genom att man dekorerar projekt rummets väggar med beskrivningar av typiska användare, arbetsuppgifter och scenarios.

Aktiv användarmedverkan i utvecklingsarbetet - de rätta användarna ska tidigt och kontinuerligt medverka aktivt under hela livscykel. Användarna bör involveras direkt i utvecklingsprojektet men även utanför själva projektets ramar. Användarna är ingen homogen grupp. Det är viktigt att se skillnaden mellan domänexperter och verkliga användare.

Evolutionär utveckling - hela utvecklingsprocessen ska bedrivas iterativt och inkrementellt. Designlösningarna bör kontinuerligt itereras med användarna. Med iterativ menas att det är fullt tillåtet och till och med rekommenderbart att man går tillbaka och modifierar det som man tidigare har utvecklat. En iteration bör innehålla följande aktiviteter: en ordentlig analys av användarnas krav och användningssammanhanget, en designfas och en dokumenterad

utvärdering med konkreta förslag till förändringar. Med inkrementell utveckling menas att systemet stegvis utvecklas uppdelat i inkrement som vart och ett levereras till verklig användning. Varje inkrement itereras tills dess uppsatta mål nåtts.

Gemensam och delad förståelse - allt ska dokumenteras så att alla inblandade förstår vad som har gjorts och vad som ska göras. Man bör använda en terminologi som är bekant för användarna. Konkreta designrepresentationer som prototyper (alltifrån enkla skisser till mer avancerade mock-up:er) och simuleringar används för att åskådliggöra den framtida användningssituationen.

Prototyping - prototyper ska användas tidigt och kontinuerligt med användare i deras arbetsmiljö för att utvärdera idéer och designlösningar. Man kan använda pappersskisser, mock-up:er och prototyper för att stödja den kreativa processen, att visualisera idéer och lösningar.

Utvärdera verklig användning - för att kontrollera att man är på rätt väg ska all design utvärderas mot de uppsatta målen. Designen utvärderas gentemot användbarhetsmål och designkriterierna tillsammans med användare, så långt som möjligt. Tidigt i utvecklingsprojektet bör man observera och mäta användarnas reaktioner på pappersskisser och mock-up:er. Senare i projektet bör användarna använda simuleringar eller prototyper för att utföra verkliga arbetsuppgifter. Deras beteende och reaktioner bör observeras, spelas in och analyseras.

Explicita och uttalade designaktiviteter - all design aktiviteter ska utföras medvetet och tillsammans med användarna. Utvecklingsprocessen skall innehålla dedikerade och medvetna designaktiviteter.

Tvärdisciplinära team - utvecklingsteamet ska ha en bredd kompetens och arbetet ska utföras effektivt av teamet. Olika kompetenser bidrar till helheten.

Användbarhetsförespråkare - personer med goda kunskaper inom användbarhet ska involveras tidigt och kontinuerligt under hela utvecklingsprocessen.

Integrerad systemdesign - alla delar som på något sätt påverkar användbarheten ska integreras med varandra för att kunna bedriva en bra systemdesign. Alla delar i systemet skall utvecklas parallellt, kontinuerligt och beroende av varandra. Detta måste ske under ansvar från en person eller grupp.

Lokalanpassa processerna - den användarcentrerade processen som projektet använder ska specificeras, anpassas och införas lokalt i varje organisation.

En användarcentrerad attityd - en användarcentrerad attityd ska alltid etableras. Det viktiga är att alla personer inblandade i projektet är medvetna om användbarhet och användarna, men i vilken grad kommer att variera beroende på projektroll och över tiden.

3.4.1 Iterativ process

Gulliksen och Göransson (2002) har beskrivit en iterativ utveckling process med tre faser: konceptuell design, interaktionsdesign och detaljerad design. Det vill säga att arbetet inleds med att identifiera behovet av användarcentrerad design. Nästa steg är målet som användarna har med systemet samt i vilken miljö systemet ska användas. Tredje steget, insamlingen av krav i en systemutveckling process, och det innebär inte att man undersöka bara funktionella kraven utan huvudpunkten ska läggas även på användbarhetskraven från användarna, därför tas det större hänsyn till användarmedverkan när man arbetar iterativt för att man går igenom serien av aktiviteter i utvecklingsprocessen flera gånger. Enligt författarna ökar möjligheterna för testning, och för att ta vara på användarna och dess arbetsuppgifter för att på så sätt utveckla en kreativ process samt medger användarna att komma med kommentarer och behov.

En iterativ utvecklings process bygger på att upprepa ett antal bestämda fasar, och då kan systemet eller den specifika system delen möta de krav på förändring som uppkommer. Arbetsflödet i en iterativ utveckling ritas oftast upp som en cirkel, och då är det viktigt att tänka på att modellen inte har något slut, då måste man planera i förväg när arbetet ska avslutas. (Gulliksen & Göransson, 2002).

3.5 Rational Unified Process

Rational Unified Process (RUP) är en utvecklingsmetod som ursprungligen kommer från utveckling och användning av ett flertal andra metoder och arbetssätt. Det var svensken Ivar

Jacobsson Ericsson som 1987 släppte Objectory Process som först utvecklades till Rational Objectory och därefter omformades tillsammans med Grady Booch och James Rumbaugh till Rational Unified Process (Lunell, 2003).

RUP är en systemutvecklingsmodell för design och implementation av mjukvaruutveckling som delar in ett projekt i fyra faser. Metoden är ett ramverk som bygger på sex ”Best Practices” (Strand, 2001), dessa beskrivs kortfattad nedan. I RUP tar man användningsfall till hjälp och processerna understryker vikten av att testa det man bygger. Användningsfallen ska följa med i hela projektet samt utvecklas och förändras kontinuerligt (Strand, 2001). Metoden kan anpassas i det oändliga, både till stora och små projekt (Strand, 2001). Tanken är att RUP ska skräddarsys så att det passar den enskilda organisationen och projektet. Utifrån detta ramverk sätter varje enskilt företag samman en process som anpassas till sammanhanget. Processerna ser därför inte ut på samma sätt i olika företag (Lunell 2003). RUP fokuserar dessutom inte enbart på produkten utan även på kvaliteten av processen (Avison & Fitzgerald, 2003).

3.5.1 Best Practice

Grundtanken i RUP är att ta tillvara gammal erfarenhet som har hämtats under årens lopp från olika håll till något som RUP betecknar som ”Best Practice”. Best Practice är metoder och principer som har arbetats fram från tidigare utvecklingsprojekt där de har identifierats som viktiga framgångsfaktorer (Lunell, 2003). Om man använder dessa metoder och principer i kombination med varandra ska man förhoppningsvis undvika de flesta problemen som kan uppstå (Kruchten, 2003). I RUPs dokumentation pekar man ut sex sådana viktiga metoder och principer vilka är följande:

1. Iterativ utveckling - en av RUPs kärnpunkter är att en programvara ska utvecklas iterativt. I det iterativa angreppssättet försöker man hitta riskerna så tidigt som möjligt i livscykeln. Därför går varje fas i RUP igenom ett antal iterationer tills man hittar så många problem det går. Fördelar med iterativ utveckling är att uppmuntra utbyte med slutanvändare så att samtliga faktiska krav identifieras efter hand (Strand, 2001).

2. Kravhantering - under en systemutvecklingsprocess ändras oundvikligen kraven under utvecklingens gång. Därför bör det finnas rutiner för hur krav och kravändringar ska hanteras. RUP processen anser därför att en systematisk kravhantering ska genomsyra hela utvecklingsprocessen.

3. Komponentbaserad utveckling - komponentbaserad utveckling är en viktig del inom mjukvaruarkitekturen eftersom den möjliggör att återanvända komponenter som redan finns tillgängliga. En komponent kan beskrivas som en del mjukvara, eller en modul som fungerar självständigt och bidrar med någon form av funktionalitet (Strand, 2001; Lunell, 2003).

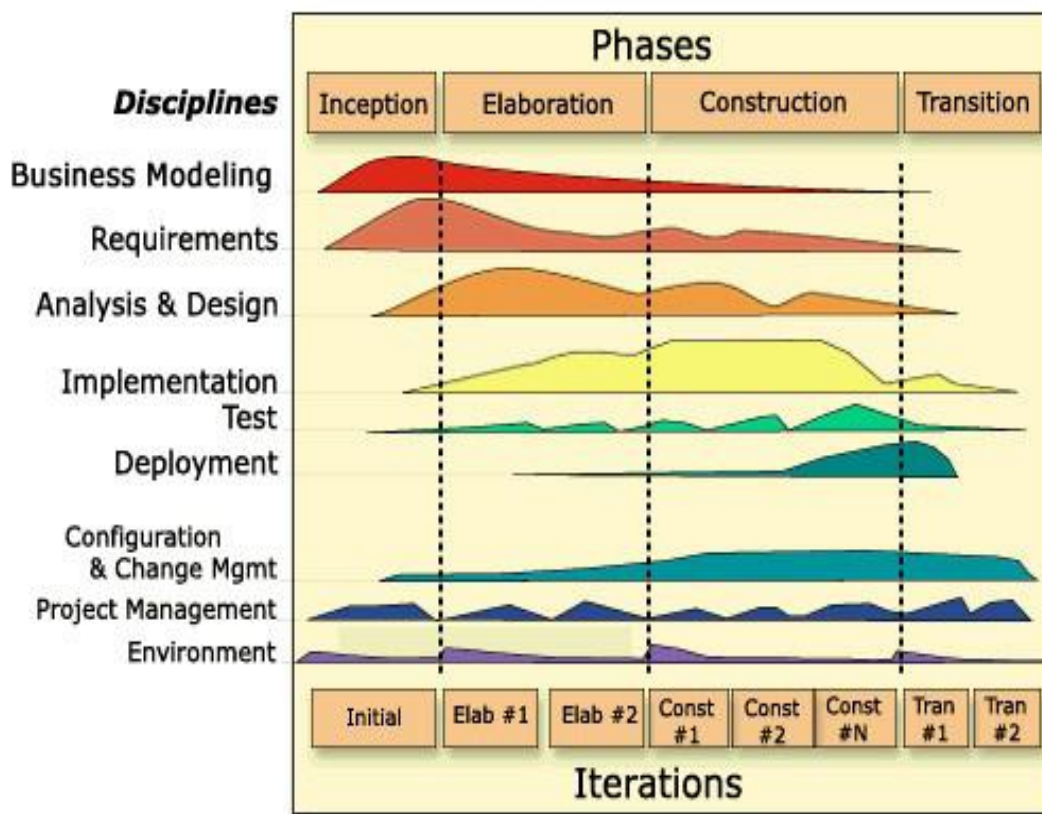
4. Visuell modellering - en modell är en beskrivning av verkligheten. Modeller hjälper utvecklare och användare att skapa en övergripande förståelse vid komplexa systemarkitekturer. Modellering är viktigt för utvecklarna då de vill få hjälp med att visualisera, konstruera, specificera och dokumentera någon del av systemet (Lunell, 2003).

5. Upprepad kvalitetsverifiering - genom upprepad kvalitetsverifiering och ett distribuerat ansvar för kvalitet kan en hög kvalitetsnivå upprätthållas i ett iterativt utvecklingsprojekt. Ett systems kvalitet definieras genom att titta på systemets tillförlitlighet, funktionalitet och prestanda (Strand L, 2001; Lunell, 2003).

6. Konfigurations- och ändringshantering - eftersom förändring är ofrånkomlig när man arbetar iterativt är förmågan att kunna hantera nya förändringar avgörande för dem som är involverade i utvecklingen. Därför finns det konfigurations- och ändringshantering (Strand, 2001).

3.5.2 Faser

RUP delar in ett projekt i fyra faser och nio arbetsflöden. Varje fas består i sin tur av ett antal *iterationer* (Strand L, 2001). En iteration är en omstart av en fas, man fortsätter faser där man slutade för att kanske lägga till nya funktioner eller hantera nya krav. Varje fas avslutas med en ”milstolpe”. Vid milstolparna avgörs projektets framtid, om det ska avslutas på grund av bristande framgång eller hinder som inte går att lösa. Visar projektet framgång, det vill säga om den håller de ekonomiska bitarna, framförallt de tidsramar som projektetplanen satt upp kommer det ett beslut om att fortsätta arbetet (Lunell, 2003).



Figur 1: Faser och discipliner/arbetsflöden (www.rational.com)

Figur 1 visar hur RUP är uppdelade i dimensionerna vertikal och horisontell. De vertikala stegen uppkallas discipliner eller arbetsflöden och de vågräta representerar faserna och iterationen i varje arbetsflöde. Faserna är olika långa och har olika arbetsbelastning och intensitet. Vidare har de också olika förutbestämde mål och syften vilka ger dem deras varierande karaktärsdrag som deras utformning och de milstolpar som skall uppfyllas. Tillsammans utgör de en utvecklingscykel och innehåller delar av en eller flera iterationer

(Kruchten, 2000). Den fas man befinner sig i kan sägas definiera projektets temporala och reella status samt även visa eventuella brister i projektet (Lunell, 2003). RUPs fyra faser är följande:

Förberedelse (*Inception*)

Etablering (*Elaboration*)

Konstruktion (*Construction*)

Överlämning (*Transition*)

3.5.2.1 Förberedelsefasen (Inception)

Under förberedelsefasen initieras projektet, syftet är i stort sett att alla parter ska vara överens om vad projektet ska åstadkomma eller utveckla (Lunell, 2003). Lunell (2003) menar att fokus ligger på att tillsammans visualisera och förstå de mest centrala kraven, sedan ska de tillsammans bilda en gemensam förståelse om vad syftet och målet är genom att skapa användningsfall. Det är viktigt att både kunden och utvecklarna kommer överens om vad som ska utvecklas. Det är inte alltid enkelt att få bilden som kunden ser framför sig stämma överens med den bild som utvecklarna får upp för sina ögon. Kunden vet inte alltid vad de vill eller vad de har för bild av systemet. Det kan ofta vara nödvändigt att göra en analys av företaget för att se vad som inte fungerar riktigt och för att hitta vart problemet ligger. Därefter få man ett godkännande av kunden för att veta att man är på rätt väg samt kunna planera resurser och arbetsinsatserna för resten av projektet. Utifrån dessa kan systemets komplexitet och omfattning identifieras och projektets risker lyftas fram. Detta arbete inriktas i huvudsak på tre områden: kravanalys, projektplanering samt utvecklingsmiljön och slutar med en milstolpe. Vidare ska användningsfallen prioriteras och projektgruppen ska bestämma vilka användningsfall som ska implementeras under nästkommande fas (Lunell, 2003).

3.5.2.2 Etableringsfasen (Elaboration)

Etableringsfasen är en fördjupning och utarbetning av det område som skisserats i föregående fas. Tyngdpunkten i denna fas flyttas till kraven och arkitekturen, ofta genom att utveckla en prototyp. Fokus ligger på att utforma och realisera de krav som har stor betydelse för systemets arkitektur. Därför är det viktigt att arkitekturen, kraven och projektplanen är stabila för att kunna ge utrymme åt förändringar. En olämplig arkitektur får som följd att begränsningar, låg effektivitet och dålig tillförlitlighet byggs in i systemet. Detta kan leda till

att produkten inte kommer att kunna svara mot förväntningarna. Felaktiga beslut kan få långtgående konsekvenser för projektet, eftersom denna fas representerar övergången från ett små sakligt till ett fullstort projekt. Här måste även kostnad och tidsåtgång kunna beräknas för hela projektet. Efter denna fas ska arkitekturen vara införd och testad vilket i sin tur innebär att de största riskerna är eliminerade (Lunell, 2003).

3.5.2.3 Konstruktionsfasen (Construction)

Under Konstruktionsfasen byggs systemet i inkrementella steg utifrån den tidigare implementerade arkitekturen samt den specificerade användarfunktionaliteten. Konstruktionsfasen delas ofta upp i flera iterationer för att få bra kontroll och styrning under utvecklingsarbetet. Systemets funktionalitet testas löpande i de olika iterationerna. Är projektgruppen uppdelade i mindre grupper som har ansvaret för att ta fram ett fåtal komponenter av systemet, arbetet effektiviseras utan att förlora i kvalitet. Innan den här fasen är över ska det finnas en färdig och godkänd betaversion som kan installeras och testas på plats hos kunden. I slutet av Konstruktionsfasen dokumenteras systemet och utbildningen planeras. Fasen slutar med en milstolpe (Lunell, 2003).

3.5.2.4 Överlämningsfasen (Transition)

Överlämningsfasen kan omfatta flera iterationer. Huvuddelen av arbetet inriktas på utvärdering och anpassning av programvaran. Fasens komplexitet har att göra med vilken typ av programvara det handlar om samt kundens framtida ansvar. Under den avslutande överlämningsfasen genomförs ett antal aktiviteter för att föra in systemet i verksamheten. Sedan genomförs bland annat installation i driftsmiljön, utbildning och acceptans. Även denna fas slutar med en milstolpe (Lunell, 2003).

3.5.3 Arbetsflöden

De fyra faserna i den horisontella dimensionen med de nio arbetsflöden i vertikala dimensionen är tillsammans schemat över hur man arbetar i RUP. Om man läser figur 1 horisontellt kan man få en uppfattning om hur arbetsinsatserna fördelar sig över tiden för ett specifikt arbetsflöde, läser man den vertikalt visar den hur en viss iteration i ett fall är uppbyggd av arbetet inom olika discipliner. Längst ner i figuren syns några exempel på

iterationer inom de olika faserna. De nio arbetsflöden är indelade i sex stycken tekniska arbetsflöden och tre stycken stödjande (Lunell, 2003). De nio arbetsflöden är följande:

Verksamhetsmodellering – Business Modeling workflow

Kravanalys – Requirements workflow

Analys & Design – Analysis & Design workflow

Implementation – Implementation workflow

Test – Test workflow

Driftsättning – Deployment workflow

Projektledning – Project management

Konfigurations och ändringshantering – Configuration and change management

Utvecklingsmiljö – Environment

3.5.3.1 Verksamhetsmodellering

Ibland behövs en modell över företaget som man utvecklar en produkt för. Det kan vara för att analysera företaget (Lunell, 2003). Lunell (2003) menar att i sådana fall kan det vara lämpligt att modellera verksamheten i stort och inte bara det datoriserade systemet som ska utvecklas. Det gör att kunden och utvecklaren får en gemensam syn på problemet.

3.5.3.2 Kravanalys

I samarbete med slutanvändaren samlas kraven in och dessa identifieras sedan. Detta är ett kontinuerligt arbetsflöde eftersom kraven förändras hela tiden. Efterhand som ett system utvecklas upptäcks nya önskemål, detta eftersom slutanvändaren hela tiden får se mer konkreta resultat av den slutliga produkten under projektets gång. Detta arbetsflöde upprepas tills alla krav är definierade och alla ändringar har hanterats, det vill säga att all respons har behandlats. Kravhantering innebär att utvärdera förändringar, bedöma vilken betydelse dessa förändringar har samt att spåra och dokumentera alla beslut angående de olika kraven. Arbetsflödet ger systemutvecklarna en bättre förståelse för systemkraven, det vill säga vad ett system skall göra och varför (Lunell, 2003).

3.5.3.3 Analys & Design

Under disciplinen, analys och design ska en beskrivning utav systemet framställas som sedan ska ligga till grund för implementeringsarbetet. Viktiga saker att tänka på är att systemet ska vara lätt att utveckla, uppdatera och förstå. Först utvecklas en analysmodell utifrån de grundläggande kraven. Denna utvecklas hela tiden under projektets gång och övergår till slut till en designmodell. Arbetet med analys och design startar redan under förberedelsefasen (Lunell, 2003).

3.5.3.4 Implementation

Här kodas och organiseras källkoden till programvaran. Som ritning för programmet har man designmodellen. Implementation har sin tyngdpunkt i konstruktionsfasen, men det är inte ovanligt att implementationsaktiviteter förekommer redan under förberedelsefasen för att prova en idé (Lunell, 2003).

3.5.3.5 Test

Här arbetar man med att säkerställa att programmet inte lider av några brister eller fel. Det finns många olika sorters tester för olika ändamål som RUP tillhandahåller. Syftet är att i möjligaste mån garantera att så blir fallet. Testningen börjar tidigt, redan under den första fasen i samband med att de gör första experimenten med design och implementering, och löper sedan parallellt med projektet ända till slutlig leverans (Lunell, 2003).

3.5.3.6 Driftsättning

Här ska produkten lämnas över till användarna för betatestning. Den ska också installeras och personalen ska utbildas. Detta startar vid slutet av den tredje fasen och i början av den fjärde, när produkten provas i sin helhet krävs det en större arbetsinsats (Lunell, 2003).

3.5.3.7 Projektledning

Det första stödjande arbetsflödet är projektledning. Detta arbetsflöde stödjer och ger råd om hur projektet ska planeras, bemannas och genomföras. Arbetsflöden hjälper även till att skapa en grund för hur arbetet med riskhantering ska genomföras (Lunell, 2003).

3.5.3.8 Konfigurations och ändringshantering

Detta arbetsflöde handlar om att se till att det finns spårbarhet bakåt, vad som har gjorts i vilka versioner och vilka versioner som är aktuella. I och med att RUP arbetar med artefakter och iterationsutveckling, kommer det att produceras många olika versioner av programvaran. Därför är det här arbetsflödet väldigt viktigt, den täcker tre av varandra beroende funktioner. Den första funktionen har att göra med identifiering av olika versioner av komponenter och vilka komponenter som är beroende av varandra. Den andra funktionen handlar om önskade förändringar någon i eller utanför projektgruppen kommit på, dels under själva förändringen men även efter förändringen. Den tredje funktionen handlar om vilken status, kvalitet och tendens en produkt har (Lunell, 2003).

3.5.3.9 Utvecklingsmiljö

Syftet med detta arbetsflöde är att stödja projektgruppen med både verktyg och processer. Arbetsflödet handlar om vilka verktyg som behövs och hur dessa ska skaffas. Det sköter även de stöd som IT kan ge, till exempel administration, infrastruktur och säkerhetskopiering. För att kunna producera bra produkter, krävs en arbetsmiljö som är tillfredställande samt att all infrastruktur för både mjukvara och hårdvara fungerar som det ska (Lunell, 2003).

3.5.4 Användningsfall

Användningsfallen illustrerar ett systems funktionalitet samt den tekniska miljön. De hjälper till att samla in krav från slutanvändare samtidigt som det fungerar som en kommunikationslänk mellan beställare, användare och utvecklingsgrupp. Användarfallen är en grundsten i RUP och modelleras genom UML. Användarfallen har som syfte att fånga användarnas krav. Vidare beskriver användarfallen systemets funktionalitet och fokuserar på specifika situationer med relevant information i förhållande till specifika användare (Avison and Fitzgerald, 2003). Det medför att användningsfall är ett viktigt underlag för aktiviteter inom analys, design och test. Användningsfall fungerar även som ett kontrakt mellan kunder och utvecklare. Den ger även möjligheter till bättre förståelse av vad systemet måste kunna klara av och skapar förutsättningar för att hantera komplexa system (Gulliksen & Göransson, 2002).

Alltså användningsfall underlättar kommunikationen i projektet samt fungerar som en häftmassa mellan krav och designaktiviteter. Användningsfallen är värdefullt i testprocessen för att den fungerar som underlag samt identifiera testfall och tesprocedurer. Detta betyder att varje användningsfall blir testat, implementerat och verifierat. Andra aktiviteter som tar användningsfallen till hjälp är de som rör dokumentationen (Strand, 2001). Är användningsfallen bra skrivna kan de återanvändas i användarstödet som består av användarmanualer, utbildningsmaterial och så vidare, eftersom de specificerar interaktionen mellan användare och system (Strand, 2001).

3.5.5 RUP är mer Agilt idag

Unified Process är en modellering som skapades av Scott Ambler som kombinerade den Rational Unified Process (RUP) till Agila metoder, då skapades en grundlig smidig process ram som kan tillämpas på alla typer av programvara projekt. Christou, Ponis & Palaiologou (2010). Agile är ett samlings namn för ett antal modeller för att utveckla mjukvara, och den betyder lättroblig. Higsmith (2001).

Grundtanken med Agila metoder är att i en föränderlig värld krävs utvecklingsmetoder som hanterar förändring som en del av verkligheten, inte sådana som blundar för förändringar eller som försöker reglera bort dem. Mats Nyman Wenell Management AB(2010).

Sutherland (2008) anser att produktivitet kan öka fem gånger som en följd av användandet av Agila metoder, för att Agila metoder dikterar att hela den traditionella livscykeln passar in i en iteration. Det vill säga att en arbetsbörda för en iteration i taget och sedan hela cykeln inträffar inom en iteration (Chiron Business Solutions, 2010).

Manifest för Agila metoder värdesätta: Individer och interaktioner framför processer och verktyg. Fungerande programvara framför omfattande dokumentation. Kundsamarbete framför kontraktsförhandling. Anpassning till förändring framför att följa en plan.

(<http://agilemanifesto.org/iso/sv/principles.html>)

De principerna som ligger bakom det Agila manifestet innehåller: Den högsta prioritet är att tillfredsställa kunden genom tidig och kontinuerlig leverans av värdefull programvara.

Välkomna förändrade krav, även sent under utvecklingen. Agila metoder utnyttjar förändring till kundens konkurrensfördel. Leverera fungerande programvara ofta, med ett par veckors till ett par månaders mellanrum, ju oftare desto bättre. Verksamhetskunniga

och utvecklare måste arbeta tillsammans dagligen under hela projektet. Bygg projekt kring motiverade individer. Ge dem den miljö och det stöd de behöver, och lita på att de får jobbet gjort. Kommunikation ansikte mot ansikte är det bästa och effektivaste sättet att förmedla information, både till och inom utvecklingsteamet. Fungerande programvara är främsta måttet på framsteg. Agila metoder verkar för uthållighet. Sponsorer, utvecklare och användare skall kunna hålla jämn utvecklingstakt under obegränsad tid. Kontinuerlig uppmärksamhet på förstklassig teknik och bra design stärker anpassningsförmågan. Enkelhet – konsten att maximera mängden arbete som inte görs – är grundläggande. Bäst arkitektur, krav och design växer fram med självorganiserande team. Med jämna mellanrum reflekterar teamet över hur det kan bli mer effektivt och justerar sitt beteende därefter.

(<http://agilemanifesto.org/iso/sv/principles.html>)

3.5.6 Tillämpning av Agila Unified Process

AgileUP är en förenklad version av Rational Unified Process (RUP) som används idag. Den är en enkel metod, och lätt att förstå. Den är smidig version av Rational Unified Process (RUP), som definierar mjukvaruleveranscykel och består av fyra delprocesser och sju arbetsflöden. Den seriella karaktär Agila UP fångas i sina fyra faser:

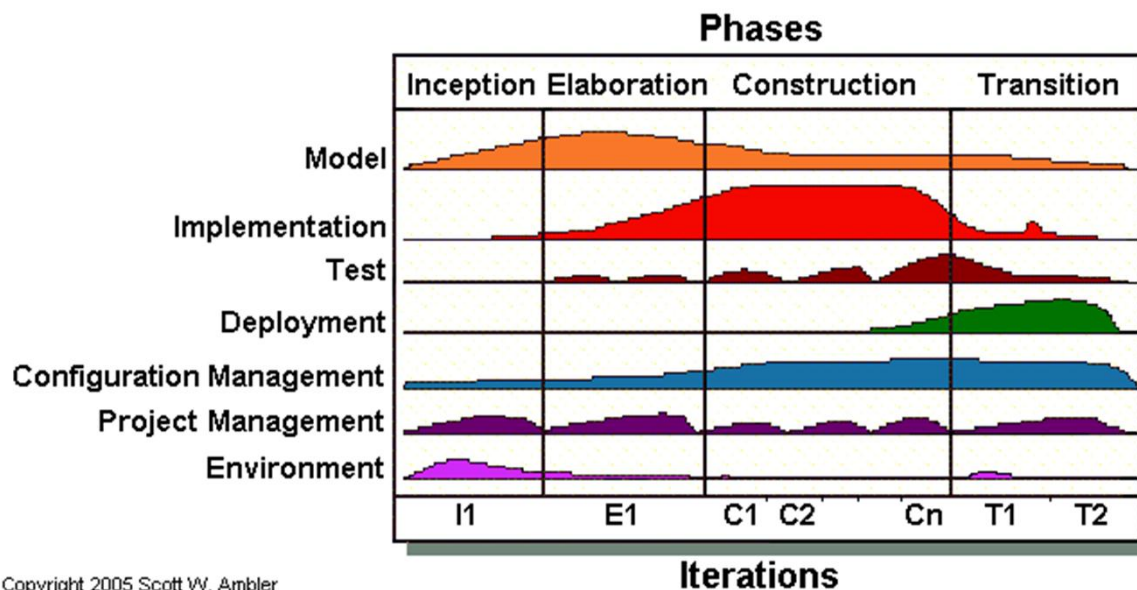
Förberedelse (Inception): Här identifieras den initiala omfattningen av projektet, en eventuell arkitektur för systemet, för att erhålla projektfinansiering och intressenter godkännande.

Etablering (Elaboration): Målet är att kunna bevisa strukturen av systemet.

Konstruktion (Construction): Målet är att utveckla inkrementell fungerande programvara som uppfyller de högst prioriterade behoven hos intressenter.

Överlämning (Transition): det gäller att validera och driftsätta systemet i produktionsmiljön.

Att sammanställa RUP och AUP arbetsflöden visar att AUP kombinerar affärsmodellering, krav, samt analys och design till en enda arbetsflöde som kallas modell (The Agila Unified Process). Flödet i AUP kräver kortare iterationer som tar ungefär två veckors iterationer som processen övergång i olika faser. Shane Hastie (2010) (The agile unified process. (n.d.). The agile unified process. <http://www.cs.sjsu.edu/~pearce/modules/lectures/se/aup.htm>).



Copyright 2005 Scott W. Ambler

Figure 1: Agile Unified Process Phases, source (Ambler, 2005)

3.5.7 Arbetsflöden

Tillskillnad från RUP, har AUP bara sju discipliner. De fyra faserna i den horisontella dimensionen med de sju arbetsflöden i vertikala dimensionen är tillsammans schemat över hur man arbetar i AUP. Ambler (2005).

Modell:

Det är första steget i AUP arbetsflöde, och målet här är att förstå verksamheten i organisationen. Den börjar med förberedelse fas. Här är identifierar en hållbar lösning för att lösa problemet, omfattning, risker, kostnader och tidsplan, och projektets genomförbarhet definieras inklusive förberedelse för projektmiljö. Ambler, (2005). Ibland behövs en modell över företaget som man utvecklar en produkt för. Det kan vara för att analysera företaget (Lunell, 2003). En viktig milstolpe i förberedelse fas är att definiera kraven utan avseende på dess slutliga tekniska lösningar. Enligt Taylor, Medvidovic & Dashofy(2010) menar " man behöver en övergripande förståelse för kraven även om man har uppfattning vad som krävs för att lösa ett problem”

Ambler (2005) förklarar att i kraven ingår identifiering av partnererna, att förstå användarens problem, lägga grund för bedömning, och definiering av användargränssnittet. Även om dessa verksamheter förekommer under Förberedelse och Etablering fas, kan de fortsätta genom andra faser också för att förbättra den pågående designen.

Ambler (2002) menar att det är viktigt att kunna beskriva funktionen på ett sätt som är oberoende av tekniken, och det kan man göra med hjälp av modeller som består av UML Case diagrams och viktiga användningsfall. Iterativt omarbete fortsätter under konstruktionsfas med user stories som förs för att återspegla den växande förståelsen av problemområdet under projektets gång (Ambler, 2005). Den sista fasen av modellarbetsflöde är överlämningsfas, även här är iterativa förbättringar som modellering, implementation, testning och förvaltningsverksamhet kan fortfarande tillämpas, innan man levereras mjukvaran till användaren. (Ambler, 2005), och (The Agila Unified Process AUP). Charles Edeki, International Journal of Computer Science and Mobile Applications (2013).

Implementation: Målet är att förvandla modellen till körbar kod och utföra en grundläggande nivå av testning. Här är viktigt med unit testning.

Test: Målet här är att göra en objektiv bedömning, och att utvärdera krav för att säkerställa kvaliteten, viktigt att hitta defekter, och systemet ska funka som avsett.

Driftsättning: Målet med denna disciplin är att planera för leverans och att verkställa planen för att göra systemet tillgängligt för slutanvändare.

Konfigurations och ändringshantering: Hantera styrningen och förändringar, och åtkomst till projektartefakter

Projektledning: Att styra den verksamhet hantera aktiviteter, risker, styra människor och vara säker på att produkten levereras i tid och inom budget.

Miljö: Målet här är att stödja och vägleda teamet och se till att rätta verktyg finns tillgängliga för alla (hårdvara, mjukvara, etc.). (The Agila Unified Process AUP). Charles Edeki, International Journal of Computer Science and Mobile Applications (2013).

När Ambler skapade AUP, var hans konstruktion inriktad kring följande principer:

- De flesta människor läsa inte detaljerad dokumentation. Utan de behöver vägledning och utbildning nu och då
- Projektet ska beskrivas helt enkelt på ett par sidor.
- Den AUP uppfyller de värden och principer som beskrivs av Agila Alliance.
- Projektet måste fokusera på att leverera väsentliga värde i stället för onödiga funktioner.
- Utvecklare ska vara fria att använda verktyg som är bäst lämpade för uppgiften, i stället för

att följa ett påbud.

– AUP är lätt anpassas via vanliga HTML-redigeringsverktyg.

(Ambler, 2005 Ambler, S. (2005). <http://www.ambysoft.com/unifiedprocess/agileUP.html>).

Charles Edeki, International Journal of Computer Science and Mobile Applications (2013)

3.5.8 RUP 2007 till 2014 – Vad har hänt?

RUP metoden har sitt ursprung från den Rational Corporation, som köptes av IBM år 2003 så nu är metoden en IBM-metodik. I RUP metoden innehåller aspekter av både iterativ och objektorienterad utveckling.(Computer Sweden). Eftersom RUP ägs numera av IBM finns det många verktyg som stödjer metoden. I 2002 skrev Philippe Kruchten, den ursprungliga arkitekten av ramverket RUP, skrev en artikel med titeln "Agility med RUP". I den uppgav han: Agility, för en utveckling av programvara, är förmågan att anpassa sig och reagera snabbt och korrekt på förändringar i sin miljö och till de kraven som följer av denna miljö. En smidig process är en som lätt omfamnar och stöder denna grad av anpassnings förmåga. Det handlar inte bara om storleken på processen eller tempot på leveransen, det är främst om flexibilitet. Han drog slutsatsen att RUP ger en ramverk som anpassar väl med agila tekniker och metoden. Shane Hastie (2010). Och det vi kan se idag att många företag har anpassat RUP till agila metoder. (Evaluating Agile and Scrum with Other Software Methodologies Mar 20, 2013)

Enligt Simon B. (2013) i dagens värld av smidig leverans har vissa programvaruteam förlorat förmågan att kommunicera vad det är de bygger och det är ingen förvåning att dessa team verkar ofta saknar tekniska ledarskap, riktning och konsekvens. Om man vill se till att hela teamet bidrar till samma mål, så alla ska kunna kommunicera effektivt vision om vad de gör. Även om de vill ha smidighet och förmågan att röra sig snabbt, måste de kunna kommunicera denna vision effektivt också. (Agile Software Architecture Sketches and NoUML apr 16, 2013). RUP är en strategi som bygger på principer för design och implementation av mjukvaruutveckling, den används ofta tillsammans med Unified Modeling Language (UML) för analys, och dokumentation av objektorienterat system.

RUP var ursprungligen avsedd att ge en ramverk för programvaruteknik. År 1997 Rational Software hade ledade en uppsättning av sju bästa praxis:

Utveckla iterativt: Utvecklas iterativt.

Hantera Krav: kravhantering ska genomsyras hela utvecklingsprocessen.

Komponentbaserad utveckling: återanvända komponenter som redan finns tillgängliga.

Visuell modellering: Användning av Unified Modeling Language (UML)

Upprepad kvalitetsverifiering: Att testa och genomföra effektiva kvalitetsstyrning på ett projekt bör vara en viktig del i varje fas av projektet från initiering till leverans.

Konfigurations och ändringshantering: Synkronisering av olika delar av systemet blir allt mer utmanande när delarna utvecklas av olika team som arbetar från olika geografiska platser med olika utvecklingsplattformar. (The Various Flavors of Unified Process by Shane Hastie on Feb 20, 2010).

RUP beskriver en generell struktur som är lämplig för alla typer och storlek på projekt, den är också lämplig för stora projekt som kan fortsätta i flera månader eller år. Dessa stora projekt är delade in i små uppgifter, var och en är en iteration som leder till en ny ökning i projektet. Den iterationen hänvisar till arbetsflödet steg för steg, tillväxten i produkt eller projekt. För ett lyckosamt system bör teamet känna igen kundernas krav och krav möjligheter.

Augustine(2005), Bashir, Salman Rizwan(2012), Kruchten(2004).

RUP representeras i två dimensioner, och dimensionerna i processen representerar milstolpar, faser och iteration i varje fas, som också kallas dynamiska aspekterna av processen.

Dimensioner är den horisontella axeln och den vertikala axeln. I den horisontella axeln är de fyra faserna som representerar processens livscykel och är baserat på de nio arbetsflöden. Den vertikala axel representerar logisk gruppering av olika aktiviteter, artefakter, discipliner, komponenter och arbetsflöden, som också kallas statiska aspekter av processen eller schemat över hur man arbetar i RUP. Bashir, Salman Rizwan(2012), Fuster, Juan(2010).

En central del av RUP är att hanterar krav, dokumentera, och organisera, men även att följa upp kraven bedöms som en viktig del i processen.

RUP delar upp projektets livscykel i fyra huvud faser och nio arbetsflöden. Varje fas består i sin tur av ett antal iterationer, och fas avslutas med en ”milstolpe”:

Förberedelse fas (*Inception*), Etablering fas (*Elaboration*), Konstruktion fas (*Construction*), och Överlämning fas (*Transition*). (The Various Flavors of Unified Process by Shane Hastie on Feb 20, 2010).

4 Teorianalys

I detta kapitel kommer jag att analysera RUPs faser utifrån användarcentrerad systemdesign för att ge min syn på hur användarmedverkan ser ut i RUP. Detta tänker jag göra utifrån de teorier som jag har beskrivit i föregående kapitel.

4.1 Förberedelsefasen

Syftet med RUPs första fas, förberedelsefasen, är enligt Shuja, Krebs (2008) att alla parter ska involveras och komma överens om vad projektet ska åstadkomma. Det vill säga att alla talar klarspråk. Fokus ligger på att utvecklarna tillsammans med slutanvändarna visualiserar och förstår kraven. Utifrån dessa kan systemets komplexitet och omfattning identifieras och projektets risker lyftas fram. Arbetet under förberedelsefasen inriktas i huvudsak på tre områden: kravanalys, projektplanering samt utvecklingsmiljön. Kravanalys är ett led i preciseringen och avgränsning av projektets storlek och omfattning. Projektplaneringen påverkas av kravbildens eftersom både ger uppfattning om projektets storlek och tips om hur utvecklingen skulle kunna planeras. I utvecklingsmiljö gäller det främst att etablera den miljö som kommer att krävas under projektets tidiga faser. Det kan även vara nödvändigt att göra en analys av företaget för att se vad som inte fungerar riktigt samt hitta var problemet ligger. Detta stämmer bra med filosofin inom användarcentrerad systemdesign, vilket säger att alla ska medverka i projektet, förstå verksamhetens mål, grunden i användarnas situation, deras uppgifter, varför och hur de utför sina uppgifter etc. (Gulliksen & Göransson, 2002). Vidare strävar användarcentrerad systemdesign efter att specificera var, när och hur användarna bör delta i utvecklingen. Användarcentrerad systemdesign betonar vikten av att möta användarna i deras egen naturliga arbetsmiljö samt identifiera lämpliga faser för användarnas deltagande och beskriva hur det skall ske. I RUPs teoribeskrivning framkommer inte tillräckligt hur, var och när användarna ska delta under denna fas, utan enbart att användarna ska vara med samt att utvecklarna ska förstå kraven. Men om detta ska ske i utvecklarnas miljö eller i användarnas miljö nämns inte i teorin.

Om vi tittar närmare på AUP. Enligt Taylor, Medvidovic & Dashofy(2010) " man behöver en övergripande förståelse för kraven även om man har uppfattning vad som krävs för att lösa ett problem" Ambler (2005) förklarar att i kraven ingår identifiering av parterna, att förstå användarens problem, lägga grund för bedömning, och definiering av användargränssnittet. Även om dessa verksamheter förekommer under Förberedelse och Etablering fas, kan de fortsätta genom andra faser också för att förbättra den pågående designen. Skillnaden med användning av Agila metoder gav bättre kontakt med användare och bättre kommunikation i teamet.

En viktig milstolpe i förberedelse fas är att definiera kraven utan avseende på dess slutliga tekniska lösningar, med detta lösningen får användarna lite mer möjlighet att vara med och förklara deras önskemål. Som Ambler (2002) menar att det är viktigt att kunna beskriva funktionen på ett sätt som är oberoende av tekniken, och det kan man göra med hjälp av modeller som består av UML Case diagrams och viktiga användningsfall, och här får användaren ännu mer möjligheter av olika modeller och prototyper. Och det kan stämma bra med användarcentrerad systemdesign som betonar vikten av att möta användarna i deras egen naturliga arbetsmiljö samt identifiera lämpliga faser för användarnas deltagande och beskriva hur det skall ske.

4.2 Etableringsfasen

Etableringsfasen är en fördjupning på RUPs första fas. Tyngdpunkten ligger på att fastställa kraven och arkitekturen som först utvecklas av en användnings falls område och sedan utvecklas till en prototyp som användaren kan testa. På detta sätt kan man fastställa att användarnas krav uppfylls. Även detta stämmer bra överens med filosofin inom användarcentrerad systemdesign, där det är viktigt att utveckla prototyper så tidigt som möjligt under utvecklingsprocessen, som sedan ska testas tillsammans med slutanvändarna. Detta ska ske i användarnas arbetsmiljö för att utvärdera idéer och designlösningar (Gulliksen & Göransson, 2002). Användarna bör använda simuleringar eller prototyper för att utföra verkliga arbetsuppgifter. Deras beteende och reaktioner bör observeras, spelas in och analyseras. Av detta får utvecklarna en utvärdering av prototypen som användes. Utvärdering av verklig användning, det vill säga mätbara mål för användbarheten och kriterier för designen är luddig och tveksam i RUP. RUP rör till begreppen iterativ och evolutionär

utveckling på ett sätt som gör att iterativ utveckling inte nödvändigtvis bedrivs inom ramen för RUP (Gulliksen & Göransson, 2002). Fokus i denna fas ligger i arkitekturen och själva systemlösningen vilket gör att fokus på slutanvändarna minskas. Detta stämmer inte överens med användarcentrad systemdesign, som säger att användare skall aktivt medverka, tidigt och kontinuerligt under hela utvecklingsprocessen.

När det gäller AUP, under etablering fasen blir det mycket vikt på att kunna bevisa strukturen av systemet, och identifiera båda kraven och partnererna, för att förstå användarens problem. Här lägger man också grund för bedömningen, och definieringen av användargränssnittet, och det gör man med hjälp av antingen modellering eller prototyper. Detta stämmer bra överens med användarcentrad systemdesign, och även med RUPs arbetssätt.

4.3 Konstruktionsfasen

Den tredje fasen, konstruktionsfasen, fokuserar på iterativa och inkrementella arbetssteg som testas löpande på prototyper i olika nivåer tills användarens krav uppnås (Lunell, 2003). Detta anser jag stämmer bra överens med användarcentrerad systemdesigns filosofi som framför iteration, dock ska iterationer enligt användarcentrerad systemdesign innehålla en ordning analys av användarnas krav och användningssammanhanget, vilket RUP saknar. Enligt användarcentrerad systemdesign skall alla delar i systemet utvecklas parallellt, kontinuerligt och beroende av varandra. Även detta stöds inte av RUP då man enbart jobbar med varje del i systemet en i taget. Enligt användarcentrerad systemdesign innebär inkrementell utveckling att systemet stegvis utvecklas, uppdelat i inkrement som vart och ett levereras till verklig användning (Gulliksen & Göransson, 2002). Varje inkrement itereras tills dess uppsatta mål uppnåts. Detta överensstämmer med teorin om RUP, vilket kallas enligt RUPs terminologi för milstolpar där man vid övergången från en fas till nästa avgör om syftet med fasen uppnåts.

Hela konstruktionsarbetet stödjer sig på den grundläggande arkitekturversionen som man tidigare arbetat fram under Etableringsfasen. Arbetet övergår från att kretsa kring analys och design samt utveckling av abstraktioner till att ta form som konkreta tillämpningar (Lunell, 2003). I RUP handlar det om att så tidigt som möjligt hitta användarkraven. Detta innebär att iterativa arbetssättet enligt användarcentraliserad systemdesign med aktiviteterna analys- designutvärdering inte följer med under hela systemutvecklingsprocessen.

I AUP så stämmer iterationer överens med användarcentraliserad systemdesign, där Iterativt omarbeta fortsätter under konstruktions fasen med user stories som förs för att återspegla den växande förståelsen av problem området under projekts gång (Ambler, 2005). Flödet i AUP kräver kortare iterationer som processen övergång i olika faser, de tar ungefär två veckor, och det betyder att användarna är med ganska mycket under de olika iterationerna. Shane Hastie (2010)

4.4 Överlämningsfasen

I den sista fasen, överlämningsfasen, är arbetet riktat på utvärdering och anpassning av programvaran. Innan produkten överlämnas är det viktigt att kontrollera att systemets kvalitetsnivå når de uppsatta målen. Här åtgärdas eventuella fel, systemets egenskaper justeras och saker som eventuellt saknas läggs till. Den viktigaste milstolpen kommer i slutet av överlämningsfasen då kontroll sker så att alla krav blivit uppfyllda samt att projektets mål och syften har uppnåtts. Vid sidan av detta arbete, måste användarna utbildas för att kunna använda produkten. Fasens komplexitet har att göra med vilken typ av programvara det handlar om och kundens framtida ansvar. Detta stämmer även med teorin om användarcentrerad systemdesign, att produkten bör avvecklas på ett snyggt och prydligt sätt. Detta bör sedan uppföljas av de projektmedlemmar genom regelbundna uppföljningsmöten, då man bland annat stämmer av projektets tidsplaner. Uppföljningen bör fokusera på återstående tid och arbete även om nedlagd tid och genomförda arbetsuppgifter ska noteras (Gulliksen & Göransson, 2002).

Den sista fasen av arbetsflöde är överlämnings fasen, även här så är iterativa förbättringar som modellering, implementation, testning och förvaltningsverksamhet kan fortfarande tillämpas, innan man levereras mjukvaran till användaren (Ambler, 2005), och (The Agila Unified Process (AUP)). Det gäller att validera och driftsätta systemet i produktionsmiljön.

4.5 Sammanfattning av kapitlet

Sammanfattningsvis anser jag att det finns hinder att bedriva användarcentrerade filosofin inom metodens olika faser och arbetsflöden. Detta på grund av att användarmedverkan och användare inte uttalas i stor utsträckning i RUP metoden. Enligt Gulliksen och Göransson (2002) övergår fokus från användarna till systemlösningen när användarkraven är fångad. Jag

håller med om detta i den meningen att användarnas medverkan minskas och typen av medverkan ändras efter att kraven är fångad.

På senare år har många företag börjat använda Agila metoder eller Agilt vid sidan om RUP riktade för utvecklings projekt. Idag finns det många olika Agila metoder som fokuserar sig på den traditionella livscykeln för att passa in i en iteration, det vill säga att en arbetsbörda för en iteration i taget och sedan hela cykeln inträffar inom en iteration (Chiron Business Solutions, 2010).

Enligt teorin anser jag att det växer fram många olika Agila metoder nuförtiden, detta har påverkat tillvägagångssättet i RUP. RUPs användning har minskat, och Agila metoder har tagit över, då blir det bättre att bedriva användarcentrerade filosofin inom AUPs olika faser, för att den är mer flexibel, enkel, och lättarbetad.

Enligt Sutherland (2008) att produktiviteten kan öka med fem gånger som en följd av användandet av Agila metoder. Därför har många företag antingen börjat med Agila metoder eller anpassat det till RUP så får de en mindre version av RUP (AUP). En blandning mellan Agila metoder och plandrivna tekniker är det allmänna för företagen just nu. Men Gibbs (2006) menar att RUP kan vara Agilt också om aktiviteterna eliminerar de processer och artefakter som inte används, och förkorta iterationer.

5 Empiri

Jag kommer i detta kapitel att presentera hur jag har utfört den empiriska undersökningen. Först ger jag en kort introduktion till hur jag har valt att redovisa det empiriska arbetet. Därefter kommer en sammanfattning av de tre intervjuer jag har gjort på de tre systemutvecklare under två tidpunkter som jag har varit i kontakt med i tre företag. Varje sammanställning inleds med en beskrivning av företaget och respondenterna. Materialet kommer dels från intervjuerna och dels från företagets hemsida. Efter det kommer en sammanställning av de tre intervjuer jag har gjort.

5.1 Undersökning

Den första undersökningen gjordes 2007 är baserad på två intervjuer av systemvetare. Svaren som erhålls från de intervjuade spelades in. Den andra undersökningen är baserad också på en intervju av en systemvetare (projektledare). Jag valde att använda intervjuer som undersöknings teknik för att jämföra ett antal begrepp som jag har undersökt. Jämförelsen fokuserar på undersöknings frågan ”I vilken omfattning stödjer RUP användarmedverkan vid två olika tidpunkter?” Detta har jag sedan transkriberat ner som text. Dessa finns bifogade som bilagor i uppsatsen (se bilaga 2). Svaren från intervjuerna kommer jag att presentera nedan, där jag först ger en beskrivning av själva företaget. Jag tycker att det är för själva uppsatsen väsentligt att ha en generell insikt om företaget. Den struktur jag har valt för empirin är att redogöra svaren i företagsvis. Jag kände att det blir mer lättläst samt man kan hålla reda på vilket företag texten handlar om.

5.2 Beskrivning av Företag 1

Företag 1 är ett konsultföretag som startades år 2000 av erfarna konsulter. Idén med företaget var att erbjuda kundanpassade utbildningsmöjligheter till de företag som krävde kompetensförstärkning. Idag har Företag 1 en stark systemutvecklingsfokus med ett brett erbjudande av IT-tjänster och produkter som konsulttjänster, utveckling och utbildning. Företaget jobbar med tre verksamhetsområden: consulting, development och education.

Organisationen bygger på samarbete och innovation. Consulting, development och education är självständiga grenar i verksamheten och det är ett gemensamt erfarenhetsutbyte som sker.

Respondenten Tobias gick ut systemvetenskapliga linjen år 2004 och startade ett eget företag samma år. Det egna företaget lyckades inte så bra och lades snabbt ner. Därefter har han i cirka ett år jobbat med produktutveckling på Företag 1 och jobbar idag som datakonsult.

5.2.1 RUP användning

Företag 1 började använda RUP när företaget startades. De plockade ut delar av metoden. Vilka delar de plockade ut var olika från projekt till projekt. Anledningen till att företaget började använda RUP är att metoden är enligt Företag 1 den mest kända och vedertagna på marknaden för systemutveckling. Det finns även en hel del färdiga verktyg till utvecklarnas förfogande. Tobias tror att ingen använder hela RUP idag eftersom den helt enkelt är för stor och omfattande för att användas i sin helhet.

I dagsläget har Företag 1 plockat ut delar från RUP och andra metoder och gjort om detta till en ny metod som de kallar RUP-light. De anser att deras projekt är så pass små att de behöver en mindre version av RUP. Vidare anser de att RUP lägger ner mycket tid på dokumentation vilket gör att metoden lämpar sig till stora projekt.

5.2.2 Respondentens syn på användare

Enligt Tobias är användarna de som kommer att använda produkten. När han talar om kunden, är det också slutanvändarna han menar. Användarna eller kunden brukar ha god kunskap om sitt område vilket underlättar arbetet.

5.2.3 Användarmedverkan i RUPs faser

Tillvägagångssättet i ett projekt är olika eftersom kundtypen är olika från projekt till projekt. Ibland är det väldigt styrt från kunden, de beskriver arbetssättet som måste följas strikt. Andra gånger handlar det om en vision som kunden berättar för dem vilket innebär att utvecklarna får jobba hårt med att fånga kraven för projektet.

Innan man börjar med förberedelsefasen sätts folk ihop från olika avdelningar till en projektgrupp. Användarna kommer in och definierar tillsammans med utvecklarna vad projektet ska innebära. Här är användarna alltid med, eftersom det är användarna som kan verksamheten. Det är även här det bestäms vad som ska göras under de olika faserna och vad som ska levereras. Efter detta sätter de igång med förberedelsefasen, där kunden presenterar kravspecifikationen. I de fall där kravspecifikationen kommer direkt från kunden är den ofta fullständig, om den inte är fullständig leder det till att kravfångsten måste utföras i större omfattning.

I förberedelsefasen är användarna med i stor utsträckning. Tobias inställning är att de ska vara med så mycket som möjligt, eftersom allt blir mycket bättre i slutändan. Känslan på vad kunden vill ha infinner sig snabbare också. Även kunden anser att det är viktigt att de medverkar vid projektets start. I detta skede försöker Tobias tillsammans med de andra utvecklarna ha med användarna från början och så mycket som möjligt för att kunna fånga in kraven på ett bra sätt. Under förberedelsefasen handlar användarmedverkan om många kontakter. I konstruktionsfasen övergår arbetet till att utveckla produkten. Tobias anser då att användarnas medverkan minskar. Han säger att under denna fas handlar användarnas medverkan om avstämningar. De kontaktar användarna när de inte förstår någon viss detalj eller saknar något. Kontakten finns hela tiden men i betydligt mindre omfattning än i den inledande fasen. Enligt Tobias lyckas de oftast få tag på en person som är med från början till slut, och som de kan vända sig till när det uppstår oklarheter. Tobias tycker att användarmedverkan i konstruktionsfasen ibland är för lite. Han anser att användarna behöver vara inblandade mer. Anledningen till att kunden inte är med så mycket i denna fas är att de har mycket annat att göra vilket gör att tiden och engagemanget inte räcker.

I den sista fasen, överlämningsfasen anser Tobias att användarnas involvering ökar något igen. Här är användarmedverkan mer intensiv jämfört med föregående fas, användarna testar systemet och kommer med förslag på förändring. Förslag på stora ändringar är svåra att genomföra i detta skede beroende på tid och kostnad.

5.2.4 RUP och användarmedverkan

Användarnas inställning varierar, en del vill och kan vara med mycket andra har varken tid eller intresse att vara med. Användarmedverkan i metoden är olika hela tiden beroende på

användarnas möjligheter och inställning till projektet. Med RUP anser Tobias att man inte pratar så mycket om användarmedverkan. Enligt Tobias ligger användarmedverkan i RUP på en låg nivå. Vidare menar han att metoden inte tvingar fram användarmedverkan så som vissa andra metoder gör. Tobias anser att ju mer användarna är med i utvecklingen desto bättre blir produkten.

5.2.5 Respondentens kritik mot RUP

Tobias tycker att kritiken mot RUP är att användarna är delaktiga i början av projektet för att sedan falla bort delvis eller helt. Detta eftersom användarna oftast har fullt upp med sitt arbete. Han anser vidare att RUP lägger ner mycket tid på dokumentation som gör att processen blir långsammare. En annan brist är att metoden är väldigt omfattande, det tar tid att sätta sig in i metoden. En fördel med RUP är enligt Tobias att om man ska göra system till andra länder som Australien och Indien till exempel, är det viktigt att man dokumenterar noga för att de ska veta exakt vad utvecklarna gör.

5.3 Beskrivning av Företag 2

Företag 2 är ett konsultföretag som hjälper kunder att få bättre effekt av sina IT-investeringar, ökad styrbarhet i sina projekt, och högre kvalitet i sina IT-system. De är en av Sveriges största specialistföretag inom användbarhet, med interaktionsdesign och expertutvärdering. Företaget erbjuder köpare och leverantörer IT-produkter konkreta åtgärder för att säkerställa användarvänlighet och verksamhetsnytta för att på så sätt öka effekten av varje investerad IT-krona. Deras kunder är bland annat AstraZeneca, IKEA, SOS Alarm, Vattenfall och Sveriges största mobiltelefonföretag.

Respondenten Erik har arbetat i IT-branschen sedan 1985. Från 1995 har han uteslutande arbetat med användbarhetsfrågor. Erik har läst till Datavetenskap i Umeå. Han har även läst MDI och kommunikationsvetenskap. Erik har bred erfarenhet från branscher som exempelvis Bank & Försäkring, Läkemedel, Media, Telekommunikation, Handel, Transport och Offentlig sektor.

5.2.1 RUP användning

Företag 2 tycker att RUP är en systemutvecklingsmetod som bestäms av den organisation eller av det projekt som ska köra RUP. De tycker att det viktigaste inte är vilken metod man använder utan att det bli ett bra resultat. Deras arbetsuppgifter är olika, om man tittar i

standard RUP har de inte speciella roller men de har två grupper: Business-analys och Business-developer. Business-analys är en form av beskrivning av ett krav där man har mer teknisk inriktning eller informations modellering. Business-developer är alltid någon från verksamheten som går ut till användarna och undersöker vad användarna vill ha. Företaget har gjort olika utökningar av RUP där de försöker beskriva vilka roller som ska finnas med. Det finns egentligen inga uttalade roller som jobbar med användbarhet.

Företaget använder sig även av en annan metod som heter Effekt Studier. Denna metod exploaterar inte RUP utan ger en systemutvecklingsprocess samt ett sätt att bedriva systemutvecklingsprocess. Den ger också en projektmetod (hur projektet styrs). Viktigaste faser som de använder mest är förberedelsefasen och etableringsfasen. Mycket av det tidiga arbetet görs i förberedelsefasen, mycket av kravarbetet, detaljer och design görs i etableringsfasen. Konstruktionsfasen är mer en fråga om att bygga och detta är ett kontrollarbete. Mycket av arbetet ligger i förberedelsefasen och etableringsfasen, mindre i konstruktionsfasen.

5.3.2 Respondentens syn på användare

Med ordet användare menar Erik de som kommer att använda sig av produkten.

Alltså slutanvändarna, därför tycker han att det är viktigt att de testat och löser problemen som de tänkte lösa, så att de kan se hur bra eller dålig produkten är.

5.3.3 Användarmedverkan i RUPs faser

Enligt Erik har inte kunden så mycket insyn i ett projekt. Det är projektledaren som rapporterar till beställare eller till en styrgrupp. Företaget har istället en businessvision som är väldigt tydligt och handlar mer om hur kunder har bestämt sig för att leda projektet och hur mycket insyn de har. Att leda ett projekt är väldigt olika eftersom kunder är olika från projekt till projekt. Han utgår från den allmänna teorin om RUP där beställare är med och beskriver Businessvision dokument. I den allmänna teorin om RUP är verksamhetsfolk ofta med som Business-developers som ska kunna beskriva kraven som anknyter till verksamheten. I teorin finns också en form av styrgrupp som definieras av RUP som kan definiera hur man ska bedriva projektet.

I RUP har man en milstolpe i varje fas, det betyder att det ska finnas viss dokumentation som är färdig för att kunna gå vidare till nästa fas. Han tycker att det är en svaghet i RUP, man kunde istället visa kunden hur långt man har kommit i produkten istället för att visa dokumentationer t.ex. ”Så långt jag har kommit eller så här har jag gjort och det här vad du ska få”. Det stora problemet med RUP är att testa på användningsfall. Detta ger inte ut mycket, samt testas inte med användare utan testas med testare. Erik menar att man ska låta användarna själva testa och lösa uppgifter som de vill lösa med produkten, för att se hur bra eller dåligt den är. Han säger att man ska skilja på metoden, det som är föreskrift av metoden, och vad som egentligen görs i verkligheten i projekten vilket inte är alltid samma. Han tycker att det bästa sättet att involvera användare är att göra skisser och prototyper för att illustrera vad de olika kraven betyder.

Det finns ingenstans beskrivit i RUP att man ska ta fram beskrivning av gränssnittet redan i kravfasen, vilket är det största problemet. Erik tycker att kunden blir involverad mest i första fasen, förberedelsefasen, och tar då första skisserna på hur det ska se ut och hur kunden förklarar sitt önskemål till utvecklare. Kunder kan inte klara av att skapa sig en bild över vad systemet ska göra, genom att läsa kraven eller användningsfall. Svagheten med RUP är att den inte tar hänsyn till kraven redan från förberedelsefasen så att man kan se hur helheten hänger ihop utan den väntar tills den kommer fram till användningsfall när allting är uppbyggt.

5.3.4 RUP och användarmedverkan

Erik anser att RUP är en systemutvecklingsmetod som bestäms av den organisationen eller av det projekt som den ska köra efter. Han anser att metoden inte pratar mycket om användarmedverkan. Även användarnas inställning varierar, en del vill vara med och har mer tid än andra. Användarna involveras inte mycket i RUP, då de inte får möjlighet att testa produkten. Metoden pekar inte ut på vilket sätt användarna ska medverka. Erik anser att RUP inte heller involverar användare samt inte gör skisser för att illustrera vad de olika kraven betyder. Brister med RUP är att metoden saknar prototyp i kravarbetet och tester av användning. Han säger att RUP påstås vara iterativ men i praktiken är den inte det. I praktiken är metoden en vattenfallsmodell med tre eller fyra faser. Det märks när man har ett projekt på

ett eller tre år, där man inte kan ändra bakomliggande faser hela tiden, vilket leder att man till slut har något annorlunda, och det man bestämde är inte giltigt längre.

5.3.5 Respondentens kritik mot RUP

Personligen tycker Erik inte om att jobba med RUP. Han tycker att alla metoder har sina för- och nackdelar. Men något bra är att man kan ändra delar i den så att metoden passar till olika projekt. Andra fördelar är att RUP är en metod som backas upp av ett antal färdiga och riktiga verktyg som är kopplat till metoden, samt att mycket av dokumentstrukturen är klar vilket gör att man slipper uppfinna den. Erik tycker även att det man behöver jobba med RUP är scenario som är beskrivning av tänkta användningsmönster. Man ska låta användarna själva lösa uppgifter med hjälp av prototyp, vilket är en iteration för att få förändringar på produkten. Detta gör inte RUP. Problemet med RUP är att metoden har ut mycket strukturer och kontroll vilket bättre passar in i stora projekt.

6 Empiri

Jag kommer i detta kapitel att presentera hur jag har utfört den andra delen av empiriska undersökningen, som gjordes 2014.

Jag har använt samma sätt att redovisa det empiriska arbetet. Därefter kommer en sammanfattning av den intervjun jag har gjort på en systemutvecklare jag har varit i kontakt med i ett företag. Sammanställning inleds med en beskrivning av företaget och respondenterna. Materialet kommer dels från intervjuerna och dels från företagets hemsida. Efter det kommer en sammanställning av den intervjun jag har gjort.

6.1 Undersökning

Den andra undersökningen som gjordes 2014 är baserad också på en intervjuer av systemutvecklare. Jag valde att använda intervjuer som undersöknings teknik för att jämföra ett antal begrepp som jag har undersökt. Med intervju menar jag personliga träffar i den meningen att intervjuaren träffar intervjupersonen och genomför intervjun. Enligt (Patel & Davidson, 2003) kan intervjuer också genomföras via telefon. Jag ställde samman 9 frågor som jag hade använt tidigare under 2007. Syftet med min frågor skulle vara att de skulle analyseras och sedan ge svar på min frågeställningar. Den Intervjuaren genomförde jag med inspelning. Intervjuaren finns bifogade som bilagor i uppsatsen (se bilaga 2). Svaren från Intervjuaren kommer jag att presentera nedan, där jag först ger en beskrivning av själva företaget.

6.2 Beskrivning av Företag 3

Företaget 3 bildades 2001 av två olika företag som slogs ihop sina avdelningar för utveckling, produktion och marknadsföring av mobiltelefoner, Företaget tillverkar mobiltelefonsprodukter. Plattformen sköter signaleringen i mobilnätet och annan grundläggande funktionalitet, och företaget lägger på de tillämpningsprogram som körs

ovanpå plattformen användargränssnittet, adressboken, javaplattformen, drivprogram för kameran, program för att spela upp ljud och video etc. (Företagets hemsida).

Företaget har använt RUP i sin utveckligns projekt tidigare men nu använder de agila metoder ganska mycket. Respondenten Tomas gick ut systemvetenskapliga linjen år 2003 och fick jobb på en liten konsult bolag samma år. Därefter fick han anställnings jobb på Företaget 3. Nu har han jobbat i cirka 6 år på Företag 3, som projektledare.

6.1.1 RUP användning

Företag 3 började använda RUP när företaget startades, men de använder Agila metoder relativt mycket nu. Enligt Tomas så har Agila metoder växt fram ganska mycket på sistone. Företag 3 plockar ut delar av RUP och använder den, och även tillämpar Agila metoder vid sidan om RUP, eller använda bara Agila UP AUP.

En projekt ledare välja ut vilka delar som känns mer relevanta för ens eget utvecklingsteam och tillämpa det med Agila metoder, att köra RUP i sin helhet för stora projekt, enligt Tomas så kräver det väldigt duktiga och erfarna projektledare.

Anledningen till att företaget började använda RUP tidigare är att metoden innehåller väldigt mycket bra idéer om hur man hanterar sina krav och hur man låter kraven styra resten av arbetet, men problemet med RUP är för mycket fokus på dokumentation, mindre fokus på kod, och användbarhet är utspritt och otydligt. Men det finns så klart många fördelar också t.ex. man kan få ett mätbart resultat så tidigt som möjligt under projektens gång, och eliminering av stora risker. Enligt Tomas som har jobbat flera år med RUP så har inga stora förändringar skett vad det gäller involvering av slutanvändare och användarmedverkan, utan den stora förändring som Tomas har upplevt är RUPs användning har minskat väldigt mycket jämfört med tidigare, och Agila metoder har tagit över, prototypen som inte täcks ordentligt av RUP har många företag börjat använda den, Tomas tycker att prototypen är viktig för utvecklingen med den kan man stämma av med användarna så att de vet vad de får innan det börjar kosta för mycket.

6.1.2 Respondentens syn på användare

Användare är de som använder systemet, de som kan mest om systemet och om behovet, han menar också att i ett IT projekt ska man inte låta tekningen styr utvecklingen utan slutanvändarna kan ta över ratten under utvecklingens gång, alla användare kan vara med och testa prototyper, resultatet blir alltid bättre och man får ett system som är användbart. Kunden

kan också vara en slutanvändare. Enligt Tomas så är det från användare ska man samla in krav och skriver ner use cases.

6.1.3 Användarmedverkan i RUPs faser

Enligt Tomas avdelning så är kunderna interna och det är de som är användarna också, de har mycket möjlighet att påverka projekten. Sättet att styra ett projekt är olika ibland beror det på storleken och tiden som ett projekt tar.

Enligt Tomas i förberedelsefasen när alla grundramarna klara från olika sektioner, då kan man titta närmare på use case, projekt plan, och initial risk i de olika faserna i processen. Företaget jobbar mycket med iterationer, varje fas genomförs i ett antal iterationer men vars antal bestämmer de inte i förväg utan det beror på tid och storlek av projektet.

I början jobbar utvecklare och testare mycket tillsammans med kunden för att sätta upp kraven, kunden ska presentera kravspecifikationen om den redan inte finns.

Enligt Tomas så är det viktigt att användarna kommer in redan i förstudiefas. Användarna är de som kan mer om funktioner därför kommer de med före IT-folket i de flesta projekten. Utvecklaren bygger själv användningsfallen av kraven och försöker stämma av det med användarna. I konstruktionsfasen så fortsätter kommunikationen mellan IT-folk och användarna, och börjar de med prototypen, men ibland vill man inte blanda in användarna för mycket i konstruktionsfasen för att det finns risk att man bli styrd.

I den sista fasen, överlämningsfasen anser Tomas att användarnas involvering fortsätter och ökar betydligt mer, jämfört med konstruktionsfasen, användarna testar prototypen och kommer med olika idéer på förändringar, därför är det viktigt att ha en klar prototyp så tidigt så möjligt.

När Tomas jobbade med RUP innan så tycker att RUP inte tillåter att användarna blir involverade på något sätt utan det är helt olika beroende på vilket projekt man kör och hur användaren blir involverad. Ibland finns det inte alls tid för att involvera de, utan istället så utvecklades systemet utifrån användningsfallen och därmed fick användaren en prototyp att testa. Tomas menar att användarmedverkan och användbarhet är något som RUP kunde fokusera mer på under konstruktionsfasen

6.1.4 RUP och användarmedverkan

Användarmedverkan betyder ju att användaren ska uppfattas som en fullvärdig medlem i arbetslaget under hela projektens tid och det är inte så i RUP. Ibland är det olika hur användarna vill ha det, Tomas tycker att det är väldigt olika från kund till kund. RUP tvingar inte användarna att vara med och påverka utvecklings process eller design utan det handlar mest om tid för båda användarna och utvecklarna.

6.1.5 Respondentens kritik mot RUP

Tomas tycker att användarna inte är med lika mycket i alla projekts faser, och det kan skapa miss nöjda användare. Han tycker också att RUP saknar ett bra stöd för återblickar, det som på engelska kallas för sprint review och därför har de börjat använda några Agila metoder vid sidan om. En nackdel till med RUP som Tomas anser är brist på kontroll pga. komplexiteten, i RUP är allt information samlat på ett ställe, det bli för mycket information och svårt att hitta det viktiga, så för att det ska vara en fördel måste man lära sig plocka ut de relevanta delarna för respektive projekt, och det kräver att man är väl utbildad och har bra erfarenhet av att arbeta med RUP.

7 Analys och diskussion

I detta kapitel sammanställas studier som gjordes 2007 på Företag 1 och Företag 2 med en ny studie som gjordes på Företag 3 från 2014, jag sammanställda kategorierna som jag har funnit från empirimaterialet. Sammanställningarna från empirimaterialet har jag relaterat till teorierna som finns i teoriavsnittet.

7.1 RUP användning

Allmänt för de tre företagen är att de använder metoden RUP i sina projekt. Respondenterna plockar ut delar av metoden som de anser passar det specifika projektet. Detta märks främst i Företag 1s fall där respondenten ser RUP som ett ramverk, där de plockar ut delar av metoden som passar ett specifikt projekt. Att RUP är ett ramverk har även Strand (2001) framhävd i sin bok. Strand skriver även att de inblandade måste ha förmågan att välja och plocka ut de delar som passar att använda i ett specifikt projekt. Vidare märks i både Företag 1s fall och Företag 2s fall att de har gjort en utökning av metoden genom att lägga till delar från andra metoder. Utifrån empirin kan jag se att i mindre projekt som Företag 1 plockas mindre bitar från RUP i ett utvecklingsprojekt, medan i mer omfattande projekt som Företag 2 används RUP mer slaviskt. I Företag 2 använder utvecklarna även en annan metod som ger ett bättre sätt att bedriva systemutvecklingsprocess och en projektmetod som innebär hur ett projekt styrs. Enligt Företag 2 garanterar inte metoden RUP att någon effekt fås. Respondenten i Företag 2 tycker att man kan effekt styra genom att komplettera olika metoder, i Företag 3 projekt ledare välja ut och tillämpa vilka delar som känns mer relevanta för ens eget utvecklingsteam och kompletteras ibland med Agila metoder.

Alla respondenterna från Företag 1, och Företag 2 tycker att man inte kan använda RUP i sin helhet eftersom metoden är alldeles omfattande.

Företaget 3 plockar de ut båda delar av metoden och anpassa det till Agila metoder. Enligt de så har RUPs användning minskat väldigt mycket på sistone och börjat med AUP. AgileUP är en förenklad version av Rational Unified Process (RUP) som används idag. Den är en enkel metod, och lätt att förstå. Den är smidig version av Rational Unified Process.

Vidare tycker alla att metoden fungerar utmärkt när man plockar ut delar ur den som behövs för ett särskilt projekt. Detta skriver Lunell (2003) och Strand (2001) att användningen av

RUP ska ske genom att välja ut de delar man anser sig ha behov av detta på grund av att metoden är alldeles omfattande för att användas i sin helhet. Utifrån detta kan jag se att det inte finns någon skillnad mellan det teoretiska och praktiska tillvägagångssättet när det gäller användning av RUP, utan skillnaden är AgileUP som är en förenklad version av Rational Unified Process (RUP) som används idag. Den är en enkel metod, och lätt att förstå.

7.2 Användarmedverkan i RUP:s faser

Sammantaget för respondenterna i de tre olika företagen är att de anser att tillvägagångssättet är olika eftersom kundtypen är olika från ett projekt till ett annat. Vidare gäller för de tre företagen att i förberedelsefasen inleds arbetet med att användarna kommer in och definierar tillsammans med utvecklarna vad projektet ska innebära. Därefter sätts arbetet igång och användarna presenterar kravspecifikationen. Respondenterna tycker att kunden blir involverad mest i förberedelse- och etableringsfasen, där man tar de första skisserna på hur projektet ska se ut tillsammans med användarna. Detta stämmer överens med Lunells (2003) teori som säger att syftet med förberedelse- och etableringsfasen är att alla partner som är involverade i projektet ska enas om vad som ska utvecklas. Det stämmer även överens med hur Gulliksen och Göransson (2002) beskriver hur användarcentrerad systemdesign ska tillämpas där användarna ska involveras tidigt i ett projekt. Respondenterna tycker att slutanvändarna ska involveras tidigt eftersom produkten blir bättre då. Detta överensstämmer med teorin om användarcentrerad systemdesign. I förberedelsefasen ser jag att användarmedverkan är mycket intensiv.

Under konstruktionsfasen är användarna inte lika involverade som under förberedelsefasen. Främsta skälet till detta är att användarna inte alltid har möjlighet att avsätta tillräckligt med tid för att kunna medverka tillräckligt i utvecklingen. Att utveckla en prototyp kan förbättra användarmedverkan enligt Företag 3, han tycker att fokus ska ligga på iterativa och inkrementella arbets steg som egentligen ska testas löpande på prototyper i olika nivåer tills användarens krav uppnås och det stämmer bra överens med Lunell (2003), men så är det inte. Enligt Företag 3 så är prototyputveckling utan RUP går fortare, om man inte är väldigt skicklig på RUP. RUP påverkar prototyputveckling mest negativt. RUP fungerar bra om man kompletterar det med Agila metoder, då får man gott om tid, tillräckligt med resurser och ekonomi. Enligt RUP så ska prototyper för användargränssnitt göras under krav disciplin,

både i Förberedelse och Etablering fasen, med syftet att hantera varierande användbarhetskrav. Prototypen för det användargränssnittet baseras på use case-modellerna, vilka har tagits fram tillsammans med slutanvändare.

Kontakten med användarna är enligt respondenten på Företag 1 mer som en avstämning. De kontaktar användarna enbart när de inte förstår något detta ser jag stämmer inte så bra överens med Lunells (2003) och Strands (2001). Alla respondenterna tycker att användarmedverkan i konstruktionsfasen är för lite. Respondenterna anser att användarna behöver vara mer inblandade även under denna fas för att produkten ska bli så bra som möjligt. I de fall där utvecklarna har svårt att ta kontakt med kunden och produkten inte blir som användarna hade förväntat sig kan utvecklarna ha kunden att skylla på. Enligt Andersens (1994) beskrivning om de yttre egenskaperna har användarna skyldighet att medverka under utvecklingen och förklara vad de vill ha.

Enligt respondenterna på Företag 1 och 2 ökar involveringen av användarna igen under den sista fasen, överlämningsfasen. Här är användarmedverkan mer intensiv, användarna testar systemet och kommer med förslag på förändring. Även här kan jag ta stöd från Andersens (1994) som skriver i sina teorier om användarnas skyldigheter om de yttre egenskaperna som användarna är expert på. Respondenterna menar att användarna har möjligheten att kunna påverka och förändra saker ända fram till detta skede, hädanefter är förslag på stora förändringar svåra att genomföra.

Enligt Företag 3 användning av Agila metoder tillåter användarna involveras lika mycket under alla faser.

7.3 RUP och användarmedverkan

Alla respondenterna anser att metoden RUP inte pratar så mycket om användarmedverkan. Vidare menar de att metoden inte tvingar fram användarmedverkan så som vissa andra metoder gör. De tycker att användarna inte involveras mycket i RUPs alla faser, då de inte får möjlighet att testa produkten och att gör skisser för att illustrera vad de olika kraven betyder. men fallet skiljer sig lite i Företag 3 där utvecklades systemet utifrån användningsfallen och användarna får en prototyp att testa, när det inte fanns tid att involvera användarna. Metoden pekar inte ut på vilket sätt användarna ska medverka. Detta har Gulliksen och Göransson (2002) menat att tankesättet inom användarcentrerad systemdesign inte går att tillämpa på

RUP, för att användarna endast involveras i första fasen av ett projekt. Därefter kan det vara svårt för användarna att klara av och skapa sig en bild över vad systemet ska göra, genom att läsa kraven eller användningsfall. Jag kan också se att respondenterna har pekat på att användarnas inställning varierar vilket kan påverka möjligheten till användarmedverkan, men metoden RUP ger inte användarna möjlighet att involveras. Detta stämmer inte heller överens med ISO-definitionen om användarcentrerad systemdesign som säger att användarna ska involveras på lämpligt nivå i alla faser (ISO 13407:1 999). Respondenten på Företag 2 säger att RUP påstås vara iterativ men i praktiken är den inte det. I praktiken är metoden en vattenfallsmodell med tre eller fyra faser. Det märks när man har ett projekt på ett eller tre år, där man inte kan ändra bakomliggande faser hela tiden, vilket leder till att man till slut har något annorlunda, och det man bestämde är inte giltigt längre.

Enligt Företag 3 skiljer sig där utvecklades systemet utifrån användningsfallen och användarna får en prototyp att testa, och komma med synpunkter och nya idéer.

7.4 Respondenternas kritik mot RUP

Enligt respondenterna är användarna involverade mycket i början av en utvecklings projekt men mindre under resten av projektet. Det främsta skälet till att slutanvändarna inte är med så mycket under etablerings- och konstruktionsfasen anser respondenterna beror på att användarna inte har tid eller intresset eller enligt Företag 3 beror på att användarna behöver lära sig mycket om teknik och design aspekten, vilket bara skulle resultera i extra jobb och extra tid för utvecklarna och ibland vill de inte blanda in användarna för mycket i konstruktionsfasen för att det finns riska att man bli styrd, samt att metoden inte ger möjlighet till att användarna ska involveras aktivt. Detta säger emot Gullikson & Göransson (2002) som skriver för att lyckas med att utveckla ett användbart system måste användarna involveras genom aktiv användarmedverkan i systemutvecklingen. Detta innebär att det finns en viss skillnad mellan det teoretiska förfarandet och den praktiska, där teorin säger att användarna ska involveras i alla faser medan det inte blir så i praktiken. Enligt respondenterna tillåter RUP användarna att medverka under hela projektet men metoden inte tvingar fram det vilket tillsammans med användarnas tidsbrist orsakar att användarna inte är med lika intensiv under mitten av utvecklingen. Enligt Företag 3 så bristen på tid att sätta sig in i RUP gör att RUP känns tungt och otympligt. Enligt respondenten som jobbar på Företag 2 har alla metoder för- och nackdelar. Respondenterna anser att nackdelarna med RUP är att metoden inte tvingar

fram användarmedverkan samt att metoden lägger ner mycket tid på dokumentation.

Fördelarna med metoden är att man kan plocka ut delar ur metoden för att få den passa till olika projekt. Andra fördelar är att metoden har ett antal färdiga verktyg som kan användas samt att en större del av dokumentationen är färdig vilket gör att man slipper uppfinna den.

Respondenten på Företag 3 anser att det har blivit bättre med att använda agila metoder eller AUP vilket gör att produktivitet och kvalitet i produkten ökar speciellt vid användning av prototyper, och det stämmer bra överens med Sutherland (2008) att produktivitet kan öka fem gånger som en följd av användandet av Agila metoder, han menar att agila metoders tekniker tillämpas för att öka utvecklingsflödet till en effektivare process.

8 Slutsats

I detta kapitel tar jag upp de slutsatser jag har kommit fram till från båda studier som gjordes 2007 och 2014, samt slutord som jag anser vara intressanta.

För denna uppsats undersökte jag under 2007 i vilken omfattning metoden RUP stödjer användarmedverkan. Detta har jag gjort genom att beskriva begreppen RUP, användarcentrerad systemdesign och användarmedverkan med hjälp av litteratur samt genom att utföra en empirisk studie i form av intervjuer med två systemutvecklare på två olika företag. Utifrån det empiriska materialet och analysen kunde jag se att de två utvecklarna jag varit i kontakt med 2007 ansåg att användarna inte involverades tillräckligt under utvecklingsprojektet. Jag drog slutsatsen att det viktigaste med användarcentraliserad systemdesign är effektiv användarmedverkan. Med detta menas att rätt användare ska medverka under hela utvecklingsprojektet. Vidare drog jag även slutsatsen från jämförelse av teorin om att RUP inte gav bra stöd för användarmedverkan samt att metoden inte ger möjlighet för utvecklarna att involvera användarna på en tillräcklig nivå. RUP tvingar inte fram användarmedverkan vilket gör att användarna inte medverkar tillräckligt i alla faser. Enligt Företag 1, ska utvecklarna i förberedelsefasen tillsammans med användarna komma överens om vad projektet ska åstadkomma. Detta stämmer bra överens med teorin om användarcentrerad systemdesign, som säger att alla ska medverka aktivt i projektet. Även respondenterna från Företag 2 framförde att användarna involveras mycket under förberedelsefasen, där utvecklarna tillsammans med användarna gör de första skisserna på hur projektet ska se ut.

Teorin om etableringsfasen och konstruktionsfasen fokuserar på arkitekturen, själva systemlösningen och iterativa och inkrementella arbetssteg vilket gör att fokus på slutanvändarna minskas. Enligt Företag 1, och 2 försvinner nästan användarna under etableringsfasen och konstruktionsfasen. Främsta skälet till detta är enligt respondenterna att användarna inte har möjlighet att avsätta tillräckligt med tid då de oftast har mycket annat att göra.

I överlämningsfasen är arbetet riktat mot utvärdering och anpassning av programvaran. Användarna involveras för att testa och kontrollera produkten. Detta stämmer med teorin om användarcentrerad systemdesign som säger att produkten ska testas tillsammans med slutanvändarna innan den överlämnas. Även respondenterna tycker att involvering av användarna intensiveras i denna fas.

Utifrån ovan nämnda slutsatser kan jag se att teorin motsvarar empirin, dvs. att det teoretiska överensstämmer med vad respondenterna tyckte om användarmedverkan i metoden.

Slutsatsen jag kan dra utifrån denna studie som gjordes 2007 och som baserat på det syfte arbetet är grundat på är att RUP inte ger bra stöd för användarmedverkan samt att metoden inte ger möjlighet för utvecklarna att involvera användarna på en tillräcklig nivå i alla faser. I förberedelsefasen involveras användare ganska mycket, men detta försvinner nästan helt under etablerings- och konstruktionsfasen då fokus ligger på systemlösning och iteration. Under överlämningsfasen ökar involveringen av användarna igen. Skälet till detta är att metoden tillåter användarna att involveras under utvecklingen men tvingar inte fram användarmedverkan vilket gör att användarna inte medverkar tillräckligt i de senare faserna.

Under 2014 gjorde jag en ny undersökning, och utifrån det empiriska materialet och analysen kunde jag se att RUP idag kan överskuggas av tillkomsten av Agila metoder, därför har jag beskrivit begreppen RUP idag, och tillämpning av Agile Unified Process, med hjälp av litteratur, samt genom att utföra en empirisk studie i form av intervjuer av en systemvetare. Enligt respondent på Företag 3 (2014) som har jobbat med RUP tidigare och jobbar med Agila metoder just nu, så har det inte skett stora förändringar under de senaste åren när det gäller slutanvändare, och användarmedverkan i RUP. Användbarheten är fortfarande utspridd och otydlig. RUPs användning har minskat och nästan upphört, många företag har börjat använda Agila metoder istället, för att de är mer flexibla, enkla, och lättarbetade. Även från litteraturdelen så kunde jag komma fram till att nuförtiden växer många Agila metoder fram, och detta har påverkat tillvägagångssättet i RUP. RUPs användning har minskat, och Agila metoder har tagit över.

Respondent på Företag 3 användning av AUP eller Agila metoder ger ett bättre stöd för slutanvändare och användarmedverkan. Både prototyping och modellering (UML) utgör stor hjälp för både slutanvändarna och utvecklarna, och även anpassbarhet och bättre effektivitet på arbetet. Det stämmer bra överens med Sutherland (2008) om att produktiviteten kan öka med fem gånger som en följd av användandet av Agila metoder, och även det stämmer

överens med användarcentrerad systemdesign som säger att prototyper ska användas tidigt och kontinuerligt med användare i deras arbetsmiljö för att utvärdera idéer och designlösningar, man kan använda pappersskisser också. Han tycker att RUP inte involverar användarna på en tillräcklig nivå i alla faser, men prototypanvändande, (som inte täcktes ordentligt av RUP innan) och de agila metodens tekniker som tillämpas för att öka utvecklingsflödet till en effektivare process, har hjälpt användare att komma igång med projekt. Enligt Företag 3 ska utvecklarna i förberedelsefasen tillsammans med användarna komma överens om vad projektet ska åstadkomma. Detta stämmer bra överens med teorin om användarcentrerad systemdesign, som säger att alla ska medverka aktivt i projektet under förberedelsefasen, där utvecklarna tillsammans med användarna gör de första skisserna på hur projektet ska se ut.

Under etableringsfasen fokuserar man på kraven och arkitekturen, och genom att utveckla en prototyp, och då blir det fokus på prototyp, och design tillsammans med användare. Även detta stämmer bra överens med teorin om användarcentrerad systemdesign som säger att designlösningar kontinuerligt bör itereras med användarna. I konstruktionsfasen utvecklas systemet utifrån arkitekturen och det blir lite involvering av användare om det inte finns svårigheter i kraven. När en prototyp är färdigimplementerad så fokuserar de mer på användare och då blir det kontinuerlig test fram till överlämningsfasen. Man kan säga att användarna är med lika mycket under alla faser. Enligt Företag 3 så jobbar de med Agila metoder och då delas projektet till sprintar (varje sprint, som är mellan 5 och 30 dagar lång) där användarna kan vara med och testa prototypen.

I överlämningsfasen ökar användarnas involvering mer, användarna testar prototypen och kommer med olika idéer på förändringar, och det kan stämma överens med teorin om användarcentrerad systemdesign, som fokuserar på användare och användbarhet genom hela utvecklingsprocessen och genom hela livscykeln.

Som sagt tidigare så är det viktigaste med användarcentraliserad systemdesign effektiv användarmedverkan under hela utvecklingsprojektet.

Utifrån ovannämnda slutsatser kan jag se från båda empirin och teorin att AUP kan ge ett bra stöd för användarmedverkan samt att metoden ger möjlighet till för utvecklarna att involvera användarna i nästan alla faser, metoden är lättanvänd, och flexibel.

Jag kan också dra slutsatsen att till skillnaden från RUP så fokuserar AUP på att tillfredsställa användare genom tidig och kontinuerlig leverans, AUP är flexibel och hanterar förändring.

En orsak att agila metoder tar över idag, kan vara att i en föränderlig värld som vi lever i idag

krävs det utvecklingsmetoder som är förändringsbara, inte sådana som blundar för förändringar eller som försöker reglera bort dem.

Jag kunde också se att en av de största skillnaderna mellan AUP och RUP är att i RUP får du för mycket och förväntas ta bort det du inte behöver, och i AUP får du för lite och förväntas lägga till det som saknas.

Slutsatsen av min undersökning är att RUP idag kan överskuggas av tillkomsten av Agil metodik i högkvalitativa mjukvaruprojekt, med korta leveranstider, och att RUP är AUP idag. Skillnaden mellan de flesta Agila metoder och RUP är att RUP fokuserar på att ta upp verktyg i processen, medan Agila är mer flexibla och ger mer frihet.

Agila UP anses vara smidiga och lättanvända, och för med sig modellering, prototyper, bättre kodkvalitet, färre buggar och en snabbare arbetsprocess för systemutvecklingsprojekt, samt den ger bra stöd för användarmedverkan samt att metoden ger möjlighet för utvecklarna att involvera användarna på en tillräcklig nivå i alla faser. RUP Användbarheten är fortfarande utspridd och otydlig, samt RUPs användning har minskat och nästan upphört.

8.1 Slut Ord

Jag har påbörjat uppsatsen år 2007, och av olika anledningar kunde jag inte slutföra den under samma år. Uppsatsen har en lång historia, men när jag tänker tillbaka i tiden så varje dag har sin egen tjusning därför väljer jag att fokusera på de sakerna, så kommer livet tillslut kännas så mycket lättare. Även i de svåraste tider som jag hade samtidigt med uppsats skrivande kan jag finna små saker som gör varje dag bättre, på det sättet får jag ork och kraft till att kämpa vidare i livet och se möjligheter än begränsningar. Uppsatsen har både ett problem och en möjlighet som min handledare sa. Självt såg jag att problemet var stort men möjligheten var större och då bestämde jag mig att göra klart den. Jag tycker det är häftigt med livet och alla val, alla möjligheter och alla utmaningar vi ställs inför. Varje dag lär vi oss något nytt i livet, och varje dag ställs man inför en ny utmaning och så går livet vidare. Idag 2014 har jag och med hjälp av min handledare kunnat slutföra min uppsats som påbörjades år 2007.

Bilaga 1

1. Hur länge har ni arbetat med RUP?

2. Dina arbetsuppgifter kopplat till RUP?

3. Använder ni enbart RUP?

Om nej, vilken/vilka andra metoder använder ni?

4. Hur stor inverkan har kunden på projektets arbete?

Om lite, varför, tycker du att det är bra?

Om mycket, varför, tycker du att det är bra?

5. Vilka av RUPs faser används i projektarbetet?

Om någon fas uteblir, vilken/vilka?

Hur påverkar detta projektet i så fall?

6. I vilka systemutvecklingsfaser är användaren inblandad?

7. Vilka fördelar och brister anser du att RUP har?

8. På vilket sätt kan användarna hjälpa till i systemutvecklingen (RUP)?

9. Anser du att användarna involveras tillräckligt?

Om de involveras tillräckligt, varför tror du det?

Om de behöver involveras mer, varför tror du det?

Bilaga 2

Intervju med Tobias

Tid: 30-35 min

Plats: Ekonomihögskolan

1. Hur länge har du arbetat med RUP?

– **Kan du börja med att berätta lite om dig själv?**

– Jag har läst systemvetenskapliga linjen på Lunds högskola mellan 2001 till 2004/2005. Jag började jobba med RUP när jag började arbetet på Företag 1.

– **Jaha så du har inte jobbat så länge med metoden RUP.**

– Nä det har jag inte. RUP inte är en metod egentligen utan det är snarare ett ramverk där man plockar ut olika delar. Vi hämtar alltid olika delar från olika metoder och gör om det till en egen metod. Så egentligen kan man aldrig säga att man arbetar med RUP.

2. Dina arbetsuppgifter kopplat till RUP?

– **Kan du beskriva eller berätta lite om dina arbetsuppgifter?**

– Våra arbetsuppgifter skiljer sig från projekt till projekt. Mycket beror på hur kunden vill ha det. Arbetsuppgifterna påverkas av de dokument som jag måste göra. Vissa delar är ett måste att göras andra delar skiljer sig från projekt till projekt. Jag följer alltid ett visst flöde.

– **När du säger kunden vem är det du menar då? Är det de som kommer att använda systemet?**

– Ja det är de som kommer att använda systemet.

– **Jag pratar mycket om användare och då menar jag också slutanvändarna, bara så att vi är överens.**

– Jaha ok

3. Användning av en annan metod.

– **Vet du hur länge företaget har använt RUP?**

– De började med att använda RUP när företaget startades, dock har de aldrig använt hela metoden utan enbart delar av metoden.

– **Jaha så företaget har aldrig använt en annan metod**

– Nej man kan inte säga att vi har använt en annan metod men de på företaget insåg tidigt att metoden är lämplig för stora projekt. Företaget driver små projekt vilket gjorde att jag

plockade godbitar från RUP och andra metoder och gjorde om det till en egen metod som jag döpte till RUP-light.

4. Kundens inverkan på projektet.

– Hur stor inverkan har kunden på projektet?

– Denna fråga är lite svårt att säga, kunden är med ganska mycket i början av ett projekt där jag sätter upp kravspecifikationen tillsammans. Men det är olika från projekt till projekt.

– Menar du att kunden är enbart med under den första fasen?

– Nej de är med sen också.

– Under vilka andra faser är de med och hur mycket?

– När kravspecifikationen och den första fasen är färdig är användarna inte med så mycket. Jag kontaktar användarna när något är oklart eller när jag inte förstår någon viss detalj. Ibland lyckas projektgruppen få tag på en person som vi kan vända oss till när oklarheter och problem uppstår.

5. Vilka av RUPs faser används i projektarbetet?

– Du sa innan att ni inte använder hela metoden. Kan du berätta om vilka faser som ni använder och hur ni använder de?

– Vad menar ni riktigt

– Vilka faser som ni använder och vad ni gör under de olika faserna

– Jag lägger ner mycket tid på den första fasen. Innan första fasen börjar sätts projektgruppen ihop. Här sätts även kravspecifikationen upp. Detta gör jag alltid tillsammans med användarna eftersom användarna är experter på sina områden. Systemutvecklarna har kunskaper inom utveckling men saknar ofta kunskaper inom till exempel kemi eller medicin, vilket underlättar om användarna är med eftersom de kan ofta sitt område bättre. Ibland kommer kravspecifikationen direkt från kunden. Denna är ofta fullständig men om den inte är det måste kravspecifikationen göras om. Efter detta går arbetet över till att jobba med själva utvecklingen. Kunden är inte med så mycket under resten av arbetet. Arbetet slutar med överlämningsfasen, där arbetet lämnas över till kunden om jag inte behöver göra ytterligare ändringar.

6. Under vilka systemutvecklingsfaser är användaren inblandad?

– Under vilka av dessa faser är kunden med?

– Under första fasen är kunden involverad till stor utsträckning. Detta tycker jag personligen är mycket bra. Ju mer kunden är involverade under denna fas desto snabbare sätts arbetet igång och slutprodukten blir mycket bättre. Om kunden deltar mycket under denna fas får jag som utvecklare också en bättre känsla om vad kunden vill ha. Under de andra faserna är kunden inte med så mycket. Under dessa faser handlar kundmedverkan om avstämningar. Jag kontaktar kunden om jag undrar över något. Men under den sista fasen, överlämningsfasen är kunden med igen. De involveras här för att de ska testa produkten och komma med förslag på förändringar. Stora ändringar är svåra att göra beroende på tid och kostnad men små förändringar brukar göras.

7. Vilka fördelar och brister anser du att RUP har?

– Generellt tycker jag att processen i kund blir långsammare med RUP. Det blir mycket dokumentation får vår sida samt mer utdragna processer som tar lång tid jämfört med vissa andra utvecklingsmetod. Denna metod har inte så mycket dokumentation, utan kör mest korta iterationer som visar kunden vad man har gjort. Detta leder till att man får snabbare feedback mellan kund och utvecklare samt får på så sätt bättre kundkontakt vilket jag tycker är långsammare med RUP. Med RUP är man noggrannare med exakt hur det ska vara och göras. Detta behöver i och för sig inte enbart vara en nackdel utan kan även vara en fördel i de fall jag ska göra system till Australien eller Indien exempelvis, vilket också är väldigt inne att göra. Eftersom kunden sitter långt bort behöver de exakt veta vad utvecklarna gör, därför är det viktigt att man dokumenterar väldigt noga. Men om kunden sitter här så tycker jag att man hellre kan prata med varandra istället för att dokumentera.

8. På vilket sätt kan användarna hjälpa till i systemutvecklingen (i RUP)?

– Vad kan användarna bidra med? Hur kan de hjälpa till?

– Användarna bidrar med att ställa kraven på systemet, vad som ska göras, när det ska göras osv. De bidrar även med själva grafiska utformningen, där har de mycket att säga till för de vet vad som ska göras först och så vidare så vi får se till att lägga upp allt i rätt ordning. Vi ställer också vissa krav för kunden som de oftast accepterar och vill ha, så det kommer nya krav och modifierar det som redan finns. De bidrar framförallt med reservkunskap, något som vi inte vet om, till exempel hur någonting fungerar som vi inte har fått med i kraven, detta kan användarna svara på för de vet exakt hur det fungerar.

9 - Anser du att användarna involveras tillräckligt?

– Tycker du att användarna involveras tillräckligt eller vill du att de ska medverka mer?

– Det är nästan alltid så att användarna medverkar för lite, det kan nästan aldrig bli för mycket. Om det blir för mycket kan vi inte få ihop någonting, men det har aldrig hänt att användarna har varit med för mycket. Jag tycker att RUP tillåter användarna att vara med men metoden tvingar inte fram det som vissa andra metoder. Metoden har iterativa tänkande och utvecklar i iterationer men har inte kunden med i alla iterationer. Anledningen till att användarna involveras alldeles för lite är att metoden lägger stor fokus på dokumentation. Det är fokus på det skriftliga som gör att man inte behöver det muntliga på samma sätt. Om man har mindre skriftligt dokumentation så tvingas man prata mer med kunden för att få den kunskap och information som man behöver. Det beror även mycket på kunden, vissa kunder är jätte glada och man ser verkligen att de vill vara med men andra kunder har varken intresse eller tid. De har annat arbete som de måste göra vilket gör att de inte har viljan att vara med. Jag tycker att desto mer kunden är med desto bättre blir produkten. Så det är viktigt att få in kunden i projektet.

Intervju med Erik på Företag 2

Tid: 45-50 min

Plats: Eriks kontor

1. Hur länge har ni arbetat med RUP?

– Jag har arbetat med RUP sen 1998.

– Jaha det var ganska länge

– Ja, jag tycker att RUP är en systemutvecklings metod som bestäms av den organisation eller av det projektet som ska köra efter RUP. Jag tycker inte om att jobba med RUP, den är krånglig och ger inte så mycket, men det viktigaste inte vilken metod man använder utan det viktigaste är att det bli ett bra resultat. Alla systemutvecklings metoder har sina för- och nackdelar.

2. Dina arbetsuppgifter kopplat till RUP?

– **Kan du berätta om dina arbetsuppgifter**

– Det är olika om man tittar i standard RUP så finns inte speciella roller men i standard RUP har man Businessanalys eller Business-developer. Business-developer är alltid någon från verksamheten som går ut till användarna och undersöker vad användarna vill ha.

Businessanalys är en form av beskrivning av ett krav som man kan ha mer en teknisk inriktning eller mer informations modellering.

– Vi har gjort olika utökningar av RUP där vi försöker beskriva vilka roller som ska finnas med, men egentligen finns inga uttalade roller som jobbar med användbarhet.

3. Användning av en annan metod.

– **Använder ni enbart RUP?**

– Nej

– **Vilka andra metoder använder ni?**

– Vi använder oss av en annan metod också som heter Effekt Studier men den metoden exploaterar inte RUP utan den ger en systemutvecklings process och ett sätt att bedriva systemutvecklings process och den ger en projektmetod (hur projektet styrs) och det säger inte RUP den garanterar inte heller att jag får någon effekt. Jag tycker också att man kan få en effektstyrning av ett projekt när man komplettera olika metoder t.ex. Projektledning.

4. Kundens inverkan på projektet.

– **Hur stor inverkan har kunden på projektarbetet?**

– Det är olika, om jag utgår från teorin så är beställaren med och beskriver Businessvision dokument, i teorin är verksamhetsfolk ofta med som Business-developers och ska kunna beskriva krav som anknyter till verksamheten. I teorin finns också en form av styrgrupp som definieras av RUP som kan definiera hur man ska bedriva projektet. Men i praktiken så har Business-development ett problem, det blir svårt att komma från verksamheten och uttrycka verksamheten av IT projekt så får man problem att fatta beslut, då kommer väldigt lite nytta ut. Det finns ingen stans beskrivit i RUP att man ska ta fram beskrivning av gränssnittet redan i kravfasen, och det är största problemen med RUP.

– **Under vilka faser är kunden med mycket?**

– Jag tycker att kunden blir involverad mest i första fasen, förberedelsefasen, och då tar man första skisserna på hur det ska ser ut och här förklarar kunden sitt önskemål till utvecklare.

– **Tycker ni att kunden ska involveras i hela projekt?**

– Utvecklarna kan inte klara av att skapa sig en bild över vad systemet ska göra, genom att läsa kraven eller användningsfall. Jag tycker att svagheten med RUP är att den inte tar hänsyn till karven redan från förberedelsefasen så att man kan se hur helheten hänger ihop utan den väntar tills den kommer fram till användningsfall när allting är uppbyggd.

– **På vilket sätt är det bäst att involvera användarna?**

– Jag tycker det bästa sättet att involvera användare är att gör skisser för att illustrera vad de olika kraven betyder, RUP pekar inte på det utan på många andra saker. Alltså alla metoder har sina för- och nackdelar, man behöver ändra delar i dem så att de passar mer till olika projekt.

5. Vilka av RUPs faser används i projektarbetet?

– De viktigaste faserna är förberedelsefasen och etableringsfasen. Mycket av tidigare arbete gör vi i förberedelsefasen. Mycket av krav arbetet och det man tar fram detaljer och design gör vi i etableringsfasen. Konstruktionsfasen är mer fråga om att bygga och detta är en kontroll arbete, men mycket av arbetet ligger i förberedelsefasen och etableringsfasen, och det är mindre i konstruktionsfasen.

– **Uteblir någon fas?**

– nej, ingen fas uteblir.

6. I vilka systemutvecklings faser är användaren inblandad?

– **Under vilka faser är kunden inblandad?**

– I projekten har inte kunder så mycket insyn, det är projektledare som rapportera till beställare eller till en styrgrupp. Istället har man Businessvision som är väldigt tydligt handlar mer om hur kunder har bestämt sig för att leda projekt och så mycket insyn de har. I RUP har man en milstolpe i varje fas och det betyder att det ska finnas visa dokumentation som är färdiga för att kunna gå vidare till nästa fas, jag tycker att det är en svaghet i RUP man kunde istället visa kunden så långt man har kommit i produkten istället att visa dokumentationer

– **Alltså på vilket sätt menar ni att man kan visa kunden?**

– T.ex. Så långt vi har kommit eller så här har vi gjort och det här är vad du ska få. Det stora problemet med RUP är att testa på användningsfall och det betyder att man testa textuella krav, detta kan inte ger ut mycket, och det testas inte med användare utan det testas som testare. Jag menar att man ska låta användarna testa själva och lösa uppgifter som de vill lösa med den här produkten, så de kan se hur bra eller dåligt den är. Man måste skilja på metoden

mellan vad det som är föreskrift om metoden, och vad som egentligen görs i verkligheten i projekten det är inte alltid samma.

7. Vilka fördelar och brister anser du att RUP har?

– Fördelar med RUP är att RUP är en metod som backas upp av antal färdiga och riktiga verktyg som kopplas till metoden, och att mycket av dokument strukturen är klar så vi slipper uppfinna det. Brister med RUP är att metoden saknar prototyp i kravarbetet, riktiga tester och med det menar jag tester av användning, RUP påstår att den är iterativ men i praktiken så är den inte så utan den är en vattenfallsmodell med tre eller fyra faser, för att när jag har ett projekt på ett eller tre år så kan man inte ändra bakomliggande faser hela tiden, till slut så har man något annorlunda, och det man bestämde är inte giltigt längre.

– Jag tycker det man behöver jobba med RUP är scenario som är beskrivning av tänkta användning mönster. Man ska låta användare själv lösa uppgifter med hjälp av prototyp och det är en iteration för att man få förändringar på produkten, och det gör inte RUP. Problem med RUP är att den lovar mycket strukturer och kontroll för att passa in nästan bara till stora organisationer som kan ta in RUP till deras stora projekt.

8. På vilket sätt kan användarna hjälpa till i systemutvecklingen (i RUP)?

– Hur kan användarna hjälpa till i systemutvecklingen?

– Det som jag har upplevt från verkligheten är referensgrupper och Business-developer, som kan ge hjälp till utvecklare. Man brukar ha en form av test där tänkta mängd av användare ska vara med, men det bli alltid försenat för att det är färdigt byggt, det påverkar inte mycket. Det finns något som heter (ändringsrecept) och den tar nästan tio timmar att administrera, och ändringen i den är dyrt.

9. Anser du att användarna involveras tillräckligt?

– Tycker du att användarna involveras tillräckligt?

– Nej, användarna involveras inte mycket i RUP för att de inte får de möjligheter att lösa tänkta uppgifter och metoden har inte pekat ut på vilket sätt man ska använda användare.

Intervju med Tomas på Företag 3

Tid: 45-50 min

Plats: Tomas kontor

1. Hur länge har du arbetat med RUP?

– **Kan du börja med att berätta lite om dig själv?**

– Jag har gått systemvetenskapliga linjen mellan 2000 till 2003.

år 2003 och fick jag jobb på en liten konsult bolag. Därefter fick jag anställnings jobb på Företaget 3. Nu har jag jobbat här i cirka 6 år. Men jag har jobbat med olika metoder inte bara RUP, nu mer så jobbar vi relativt mycket Agilt vid sidan om RUP eller vi kan säga AUP.

2. Dina arbetsuppgifter kopplat till RUP/AUP?

– **Kan du beskriva eller berätta lite om dina arbetsuppgifter?**

Ja, som projektledare så har jag hög grad av ansvar och styr hela systemutvecklingsprojektet.

– Våra arbetsuppgifter skiljer sig från projekt till projekt. Mycket beror på hur kunden vill ha det. Arbetsuppgifterna påverkas av de dokument som jag måste göra. Vissa delar är ett måste att göras andra delar skiljer sig från projekt till projekt. Jag följer alltid ett visst flöde.

– **När du säger kunden vem är det du menar då? Är det de som kommer att använda systemet?**

– Jag menar de som använder systemet.

– **Jag pratar mycket om användare och då menar jag också slutanvändarna, bara så att vi är överens.**

– ok

3. Användning av en annan metod.

– **Vet du hur länge företaget hade använt RUP?**

– vi började med RUP när företaget startades. Tidigare har vi aldrig använt en annan metod än RUP, men nu har använder vi Agila metoder ganska mycket vid sida om RUP eller själva till båda stora och små projekt, Agile UP är mycket lättare, man känner sig mer flexible, och kan använda alla verktyg. Man behöver köra båda plandrivna arbetsätt och Agila metoder, med agila metoden bryter vi ner utvecklingsgången till mindre delar och bestämmer vad som ska göras under den nästa skedet.

RUP är stort ramverk och kräver utbildningar, stora dokumentationer, och mycket erfarenhet, men RUP hade växt fram och varit den bästa metod med värdefulla strategier tidigare, men nu på sistone så har användning av RUP minskat jätte mycket.

4. Kundens inverkan på projektet

– Hur stor inverkan har kunden på projektet?

RUP är ju ett ramverk som är iterativ och ingen metod, den fungerar bäst om det finns en aktiv kommunikation med kund, och därför är kunden med ganska mycket i början av ett projekt där vi sätter upp kravspecifikationen tillsammans. Men det är olika från projekt till projekt. Nu kör vi mycket Agilt och kunden är med ganska mycket speciellt när vi får en färdig prototyp, prototyper ska användas så tidigt så möjligt och låta användare utvärdera och komma fram med idéer och designlösningar tidigt i projekt, men tyvärr så är inte fallet med RUP, det inte finns stadgar eller riktlinje som tvinga fram användare att vara med under projektets gång.

– Menar du att kunden är enbart med under den första fasen?

– Nej; det är inte helt riktigt. Innan vi börja med prototyper, och Agila metoder så var de med bara i början, nu är fallet mycket bättre, men att ha de med på alla faser nästan omöjligt med tanke på projektens tid, men det är mycket bättre när det är stora projekt då har man mer tid. Prototyputveckling utan RUP går fortare, om man inte är väldigt duktig på RUP. RUP påverkar prototyputveckling mest negativt eftersom, så många inte kan det. RUP fungerar bra om man komplettera den med Agila metoder, gott om tid, resurser och ekonomi.

– Under vilka andra faser är de med och hur mycket?

– I förberedelse fasen brukar användarna vara med ganska mycket, men när kraven och granskning är färdig då börjar bli lite mindre. Under hela projektens tid så brukar vi ha en kontakt person, i fall vi får frågor eller funderingar ang. kraven.

I etablering fokuserar vi oss på kraven och arkitekturen, och genom att utveckla en prototyp, och då blir det fokus på användare också. I konstruktionsfasen utvecklas systemet utifrån arkitekturen och det blir inblandning av användare men i mindre skala om det inte finns svårighet i krav, när en prototyp är färdig implementerat så fokuserar vi mer på användare och då blir det kontinuerlig test fram till överlämningsfasen, man kan säga att de är med under

nästan alla faser med tanke att vi jobbar Agilt nu så har vi två veckors sprint möte och då kan användarna vara med och testa prototypen.

5. Vilka faser används i projektarbetet?

– Kan du berätta om vilka faser som ni använder och hur ni använder de?

– Vi gå genom alla fasar, börjar alltid med förberedelse fasen där är löpande utvärdering och granskning, lära känna kunder, vilka projektmedlemmar som ska vara med, granska risker och hinder, och viktigaste att hantera kraven, och det gör vi med kunden, bra och fungerande dokumentation. Etablering Konstruktion och avsluta alltid med överlämning där arbetet lämnas över till kunden. Att använda RUP kräver det en erfaren projektledare som har planerat tidigare projekt med RUP, men det går mycket enklare att använda AUP nu.

6. Under vilka systemutvecklingsfaser är användaren inblandad?

– Under vilka av dessa faser är kunden med?

– I RUP så tycker jag att det är mest under förberedelse fasen är kunden involverad till stor omfattning. I början jobbar utvecklarna och testare mycket tillsammans med kunden för att sätta upp kraven, I konstruktionsfasen så fortsätter kommunikationen mellan oss och användarna, men mindre än förberedelsefasen, för att det kräver tid att lära dem, och det blir extra jobb och extra tid för utvecklarna och ibland vill man inte blanda in användarna för mycket i konstruktionsfasen för att det finns riska att man bli styrd. RUP tillåter inte att användarna bli involverade på något sätt utan det är helt olika från projekt till projekt hur användarna är involverade, och det handlar mest om tid, istället så utvecklades systemet utifrån användningsfallen och användarna får en prototyp att testa. Jag tycker att användarmedverkan och användbarhet är något som RUP kunde fokusera mer under konstruktionsfasen. I den sista fasen, överlämningsfasen användarnas involvering ökar mer. Med Agila metoder så testar användarna prototypen så tidigt så möjligt i projekt, och kommer med olika idéer på förändringar, men det bli så klart svårt och göra stora ändringar i så sent skede.

7. Vilka fördelar och brister anser du att RUP har?

– Jag tycker att användarna inte är med lika mycket i alla projektens faser, och användbarhet är utspritt och otydligt och det kan skapa miss nöjda användare. En nackdel till med RUP är bristen på kontroll pga. komplexiteten, och bristen på tid att sätta sig in i RUP gör att RUP känns tungt och otympligt. Man känner sig bunden till några verktyg.

Jag tycker att RUP saknar ett bra stöd för återblickar, det som på engelska kallas för sprint review och därför har vi börjat använda några Agila metoder vid sidan om. I RUP är allt information samlat på ett ställe det bli för mycket information och svårt att hitta det viktiga, så för att det ska vara en fördel måste man lära sig plocka ut de relevanta delarna för respektive projekt, och det kräver att man är väl utbildad och har bra erfarenhet av att arbeta med RUP. En fördel med RUP t.ex. man kan få ett mätbart resultat så tidigt som möjligt under projektets gång, och eliminering av stora risker, att metoden innehåller väldigt mycket bra idéer om hur man hanterar sina krav och hur man låter kraven styra resten av arbetet, men ibland bli det för mycket fokus på dokumentation, och mindre fokus på kod.

8. På vilket sätt kan användarna hjälpa till i systemutvecklingen?

– Vad kan användarna bidra med? Hur kan de hjälpa till?

– Det är så klart att komma med ett tydligt krav, för att det är de som kan mest om systemet och om behovet, i ett IT projekt ska man egentligen inte låta tekniken styra hela utvecklingen utan slutanvändarna kan ta över ratten under utvecklingens gång, alla användare kan vara med och testa prototyper, resultatet blir alltid bättre och man får ett system som är användbart. Och det är från användare samlar man use cases.

9 - Anser du att användarna involveras tillräckligt nu?

– Tycker du att användarna involveras tillräckligt eller vill du att de ska medverka mer?

– Användarna involveras alldeles för lite innan med RUP, men med hjälp av Agila metoder och prototyp blir det bättre. Innan vi börja med prototyper så var de med bara i början, nu är fallet mycket bättre, användarna involverar nästan lika mycket på alla faser, men med tanke på projektens tid kan påverka det ibland. Vill man lyckas med en användbar produkt bör användarna medverka aktivt under hela utvecklingsprocessen. som jag sa tidigare att det är de som kan mest om produkten, men tyvärr problemet med RUP att den innehåller inte stadgar eller riktlinje som tvinga fram användare att vara med under projektets gång.

Men det har blivit mycket bättre med Agila metoder tycker jag.

Referenser

Litteratur

Ambler, S. (2005). *The agile unified process (aup)*.

Ahmad K Shuja, Jochen Krebs (2008) *IBM Rational Unified Process Reference and Certification Guide: Solution Designer (RUP)*

Andersen, E. (1994). *Systemutveckling - principer, metoder och tekniker*.

Avison, D. & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques and Tools*.

Backman, J. (1998). *Rapporter och uppsatser*.

Bryman, A. (2002). *Samhällsvetenskapliga metoder*. Liber förlag, Malmö

Bryman, A. (2011). *Samhällsvetenskapliga metoder*. Liber förlag, Malmö

Denscombe, M. (2000). *Forskningshandboken - för småskaliga forskningsprojekt inom samhällsvetenskaperna*.

Easterby-Smith, M. R. Thorpe & A. Lowe (2002) *the philosophy of research Design*. Sage Publications, London

Flowerdew, R. & Martin, D. (2005). *Methods in human geography*. Bell & Baim, Glasgow

Gulliksen, J. & Göransson, B. (2002). *Användarcentrerad systemdesign*

Higsmith, J. (2001). *History: The Agile Manifesto*

- Holme, L. & Solvang, B. (1997). *Forskningsmetodik: om kvalitativa och kvantitativa metoder*. Studentlitteratur, Lund
- Kniberg .H. & Skarin .M. (2010) *Kanban and Scrum - Making the most of both*: Lulu.com
- Kruchten, P. (2000). *The Rational Unified Process – An Introduction*. Addison-Wesley, New York
- Kruchten, P. (2003). *The Rational Unified Process: An Introduction (3rd Edition)*
- Lunell, H. (2003). *Fyra rundor med RUP*. Studentlitteratur, Lund
- Löwgren, J. & Stolterman, E. (2004). *Design av informationsteknik - materialet utan egenskaper*.
- Gibbs, R. Dennis (2006). *Project Management with the IBM Rational Unified Process: Lessons from the Trenches*. IBM Press
- Marklund, S. (2000). *Arbetsliv och Hälsa 2000*. Arbetslivsinstitutet, Stockholm
- Merriam, B. S. (1994). *Fallstudien som forskningsmetod*.
- Ottersten, I. & Berndtsson, J. (2002). *Användbarhet i praktiken*.
- Patel, R. & Davidsson, B. (2003). *Forskningsmetodikens grunder: att planera, genomföra och rapportera en undersökning*.
- Per Kroll, Bruce MacIsaac (2006) *Agility and Discipline Made Easy: Practices from OpenUP and RUP*.
- Svenning, C. (2003). *Metodboken*.
- Stone et al, (2005) *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*.

Taylor , Medvidovic & Dashofy (2010)*Software architecture foundations, theory, and practice*

Artikel:

Augustine, Sanjiv. 2005. Managing Agile Projects. Prentice Hall Professional Technical Reference

Bashir, M. Salman and M. Rizwan Jameel Qureshi. 2012. "Hybrid software development approach for small to medium scale projects: RUP, XP and Scrum." Sci.Int. (Lahore) 24(4): 381-384.

Charles Edeki (September- 2013) International Journal of Computer Science and Mobile Applications,
Vol.1 Issue. 3, pg. 13-17 ISSN: 2321-8363
Agile Unified Process (http://www.academia.edu/4592144/Agile_Unified_Process)

The agile unified process. (n.d.). The agile unified process.

<http://www.ambyssoft.com/unifiedprocess/rupIntroduction.html>

Chiron Business Solutions, (2010) LLC: What is the difference between RUP and SCRUM methodologies

Development Teams Rational Software White Paper
TP026B, Rev 11/01

Fuster, Joan Escamilla and Juan Carlos Ruiz Garca. 2009/2010. Modern Software Project Management. University Politecnica De Valencia, Helsingin Yliopisto

John Smith A Comparison of RUP and XP.

How to Fail with the Rational Unified Process: Seven Steps to Pain and Suffering

Mats Nyman Wenell Management AB(2010)

Murray Canto. Rational Unified Process for Systems Engineering
Part 1: Introducing RUP SE Version 2.0

Shane Hastie (Feb 20, 2010) The Various Flavors of Unified Process.

Rational Unified Process Best Practices for Software.

<http://www.cs.sjsu.edu/~pearce/modules/lectures/se/aup.htm>

My-Project-Management-Expert.com (2009)

<http://kravbloggen.se>

<http://www.softdevarticles.com>

<http://www.rational.com>

<http://www-01.ibm.com/software/rational/>

<http://www.infoq.com/news/2010/02/flavors-of-unified-process>

<http://www.chiron-solutions.com/chiron-professional-journal/2010/12/20/what-is-the-difference-between-rup-and-scrum-methodologies/>