

Återanvändbara Use Cases inom telekomindustrin

Kandidatuppsats, 10 poäng, inom systemvetenskapliga programmet

Framlagd: Oktober 2006

Författare: Jens Frost

Tomas Hellström

Handledare: Hans Lundin

Examinator: Anders Svensson, Claus Persson

Återanvändbara Use Cases inom telekomindustrin

Jens Frost, Tomas Hellström

Kandidatuppsats, 10, poäng

Omfång: 57 sidor

Handledare: Hans Lundin

Resumé

Syftet med uppsatsen var att bidra med kunskap om hur Use Cases kan användas och om det fanns en möjlighet att återanvända dem. Use Cases är ett av de mest använda verktygen för modellering, de beskriver en sekvens av en händelse som inskaffar ett mätbart värde. Use Cases fångar funktionaliteten i olika situationer. Vår studie var koncentrerad på skillnaden mellan Use Cases så som vi uppfattat Rational Unified Process (RUP) modell jämfört med Telecas återanvändbara Use Cases. Undersökningen gjordes på telekomföretaget Teleca. Där gjorde vi observationer och intervjuer angående Use Cases. Teleca gick igenom effektiviteten med återanvändningsbara Use Cases och även arkiveringen och spårbarheten med återanvändningsbara Use Cases. Viktigt är att det fungerar annars kan det vara svårt att återanvända Use Cases.

Tack till alla involverade

Vi vill tacka alla som har varit inblandade i denna uppsats. Ett stort tack till Teleca och dess personal som ställt upp för oss. Deras kunskap om Use Cases har varit till stor hjälp för oss i vårt arbete. Ett speciellt tack till Robert Klang som har lagt ner mycket tid för att visa oss hur Use Cases fungerar på Teleca. Han har hjälpt oss att förstå hur Use Cases används på Teleca. Detta trots att hans tid varit begränsad, som projektledare för det projektet vi var involverade i. Vi vill även tacka Alfa Laval Lund AB och Patrik Östberg Application Manager. Tack ännu en gång för Er hjälp!

Nyckelord

Återanvändbarhet, Use Case, RUP

Källförteckning:

1	Inledning	5
1.1	Avgränsningar för litteraturstudien.....	6
1.2	Avgränsningar för projektet.....	7
2	Metod och undersökning	8
2.1	Kunskapsprocessen för uppsatsen.....	8
2.2	Urvalsförfarande	9
2.3	Den litterära studien.....	9
2.4	Observationer	10
2.4.1	<i>Frågeställningar med Quick and Dirty</i>	11
2.4.2	<i>Ostrukturerade intervjuer</i>	12
2.4.3	<i>Intervjuschemat, visuella ramverket och kodningsramens utseende</i>	13
2.5	Våra beslut och val.....	15
3	Litteraturgenomgång.....	18
3.1	RUP:s och Use Cases historia.....	18
3.1.1	<i>Vad är RUP</i>	19
3.2	Vad är Use Cases	19
3.2.1	<i>Möjligheterna till återanvändning av Use Cases idag</i>	22
3.2.2	<i>Nästa generations Use Cases</i>	22
3.2.3	<i>Användningsområden</i>	23
3.3	Viktiga begrepp i Use Cases	23
3.3.1	<i>Grundkrav inom Use Cases</i>	24
3.3.2	<i>Use Cases uppbyggnadsdelar</i>	25
3.4	Vad menas med konceptet spårbarhet?.....	29
4	Telecas form av Use Cases	32
4.1	Teleca som företag	32
4.2	Modulverksamhet	33
4.3	Telecas syn på möjligheten att göra ändringar på Use Cases	34
4.4	Telecas Use Cases.....	35
5	Diskussioner och resultat	37
5.1	Vår diskussionsgrund.....	37
5.2	Användningsområden för återanvändbara Use Cases.....	39
5.3	Skillnader mellan viktiga begrepp i ett normalt och ett återanvändningsbart Use Case.....	39
5.3.1	<i>Use Cases olika delar</i>	41
5.4	Vikten med spårbarhet inom Use Cases	42
5.5	Vikten av arkivering av Use Cases	43
5.6	Vinster vid användande av återanvändningsbara Use Cases	43
5.7	Summering av skillnader och likheter	44
6	Slutsatser.....	47
6.1	Vidare forskning	48
7	Referenslista	49

Figurer och tabeller:

Figur 2-1.....	8
Tabell 2-1	15
Figur 3-1.....	21
Figur 3-2.....	26
Tabell 3-1	27
Figur 3-3.....	30
Tabell 5-1	44

Bilagor

- Akronymer och förkortningar
- Business Process Model
- Reusable Use Case image Teleca
- Use Case image Teleca

1 Inledning

Uppsatsen bygger på en motsägelse så som vi har uppfattat den. Nämligen om hur Use Cases används och hur det arbetas med dem, gentemot den litterära uppfattningen av hur Use Cases kan och ska användas. Vi har undersökt den praktiska motsägelsen genom ett företag som producerar komponenter till den mobila telefonmarknaden där det ständigt finns återkommande krav och moduler i deras tillverkning.

Use Cases beskrivs som en sekvens av en händelse som införskaffar ett mätbart värde. Inom problemsökning finns ett sådant område där det kan behövas värden som går att mäta, exempelvis när det formuleras representativa användarscenarier. Use Cases fångar situationer mellan intressenter och systemet samt om dess sätt att kunna fungera. Det beskriver olika förutsättningar om hur systemet kan svara mot kraven från sina intressenter. Det finns olika situationer av sekvenser eller scenarier. Dessa kan uppvisa beroenden av det specifika kravet mot förhållanden för andra krav som finns runt systemet (Cockburn, 2001).

I den teoretiska delen har vi studerat litteratur från de som skapade ramverket Rational unified process(RUP), vilka beskriver varje Use Case som en unik händelse och icke återanvändningsbart. Skapandet av RUP utfördes av Ivar Jacobson, Grady Booch och James Rumbaugh samt deras team. De såg att det finns möjligheter att ändra på det unika och i dags datum håller de på att vidareutveckla metoden återanvändning av Use Cases under International Business Machines (IBM) softwares regi, som är ägare av Rational Unified Process (Jacobson, 2006).

Problemet med återanvändning är att Use Cases ofta beskrivs som unika händelser vilket gör dem unika och därmed inte möjliga att återanvända. Tekniken utvecklas och det medför att det kanske går att negligera att de ska vara unika. Vi anser att möjligheten att återanvända Use Cases skulle vara en stor fördel i dagens IT-programmeringssamhälle. Där tittas det mycket på livscykelmodellen vilken bygger på att ett system har en begränsad livslängd och att det under resans gång kan gå att återanvända delar av systemet för att förlänga livslängden på systemet.

Vår start för detta projekt började med att vi undersökte området om Use Cases som vi fann intressant för oss.

Vi har uppfattat Use Cases som ett av de mest använda verktygen för modellering, vilket vi finner intressant. Vi studerade till en början ytligt litteratur om Use Cases för att se vad det innebär, för att sen fördjupa oss inom området både genom litteraturen och via observationer. Det framkom då att vinklingen om återanvändbara Use Cases är ett hett område. Det kändes som att det fanns anledning att titta närmare på det område som vi

ville undersöka. För att riktigt fördjupa oss inom området krävdes det en djupare litteraturstudie om Use Cases, samt om själva ramverket RUP.

Vi fann vissa iakttagelser som var intressanta, så som varför Use Cases inom RUP bör vara unika i sina händelser. Det medför att Use Cases i sig själva bör vara unika, även om utvecklingen idag går mot återanvändbarhet (Jacobson, 2006). En annan orsak att studera litteraturen var att lägga fram belägg för det som vi hoppas ska komma fram i vår uppsats. Detta gjordes genom att gå igenom material från olika håll och vinklingar inom litteraturen. Vi fann litteratur som t.ex. ”C++ How to program” (Deitel, Deitel, 2001) och ”Interaction design” (Preece et al., 2002) där Use Cases togs upp. Även mer specialiserad litteratur som ”The rational unified process an introduction: Third edition” (Kruchten, 2003) och ”A Use Case driven approach” (Jacobson et al., 1993) fann vi. Dessa är axplock av den litteratur som använts.

Anledningen till att vi gjorde vår undersökning på Teleca var att vi, efter en förfrågan angående hur de hanterar Use Cases, blev erbjudna att få delta i ett projekt inom företaget. Detta för att kunna se om det fanns möjligheter för dem att vidareutveckla deras tankar om att gå vidare med att återanvända Use Cases

Anledningen till att vi gjorde vår undersökning på Teleca var att vi blev erbjudna att få delta i ett projekt inom företaget. Detta för att kunna se om det fanns möjligheter för dem att vidareutveckla sina tankar om att gå vidare med att återanvända Use Cases. Projektet gick delvis ut på att återanvända Use Cases och vi fick möjlighet att prova på detta inom ramverket RUP. Tilläggas skall att företaget har börjat arbeta med Use Cases och söker efter en övergång till att kunna återanvända Use Cases på en standardiserad basis.

För vår del så satte detta igång en dokumentstudie av materialet från projektet på Teleca tillsammans med materialet från vår litteraturstudie. Genom detta fick vi möjligheten att se om det fanns möjligheter att återanvända Use Cases inom telekomindustrin.

Syftet med uppsatsen är: Att bidra med kunskap om hur Use Cases kan användas och om det finns en möjlighet att återanvända dem.

Detta leder oss till vår forskningsfråga vilken är: Går det att återanvända Use Cases på ett företag inom telekomindustrin?

1.1 Avgränsningar för litteraturstudien

Vi har valt att göra vissa avgränsningar för att tydliggöra vilket målet med uppsatsen är så att det inte används för mycket irrelevant material, vilket kan leda till blinda hörn. Vi vill samtidigt ändå se till att den är rik på information. För att uppnå detta var vi tvungna att sätta upp gränser, annars var risken stor att uppsatsen hade växt till oproportionella mått. I vårt fall var det så att Use Cases kommer i många olika former och att de används inom många olika områden att vi insåg att vi inte kunde täcka alla dessa områden. Ett exempel på det är Essentials Use Cases som har en lite annorlunda vinkling till hur krav hittas. De

skiljer sig en del från den grafiska notationen till skillnad från de flöden som finns inom RUP. Användaren har bland annat sin interaktion i en riktning medan systemet svarar användaren med en annan aktion. Ur detta hittas de behov som är nödvändiga på olika sätt. Det är ett av skälen till att vi har valt att inte använda oss av några andra Use Cases än de som finns beskrivna i RUP. Detta var samtidigt för att komma nära de Use Cases de använder på företaget Teleca, där man söker efter en möjlighet att kunna återanvända Use Cases. Anledningen till att vi ville ha Use Cases som är lika varandra var för att jämförelsen vi gjorde skulle vara på företeelser som löser samma problem. Dessutom så ville vi vara säkra på att de Use Cases som används ska följa RUP:s standard. Vår litteraturstudie är därför koncentrerad på Use Cases som vi uppfattat Ivar Jacobsons uppbyggnad och RUP-modellen i sin helhet. Vi har valt att avgränsa oss mot att inte ta upp själva implementeringen av Use Cases, dvs. programmeringen av mjukvaran som skapas med hjälp av Use Cases.

1.2 Avgränsningar för projektet

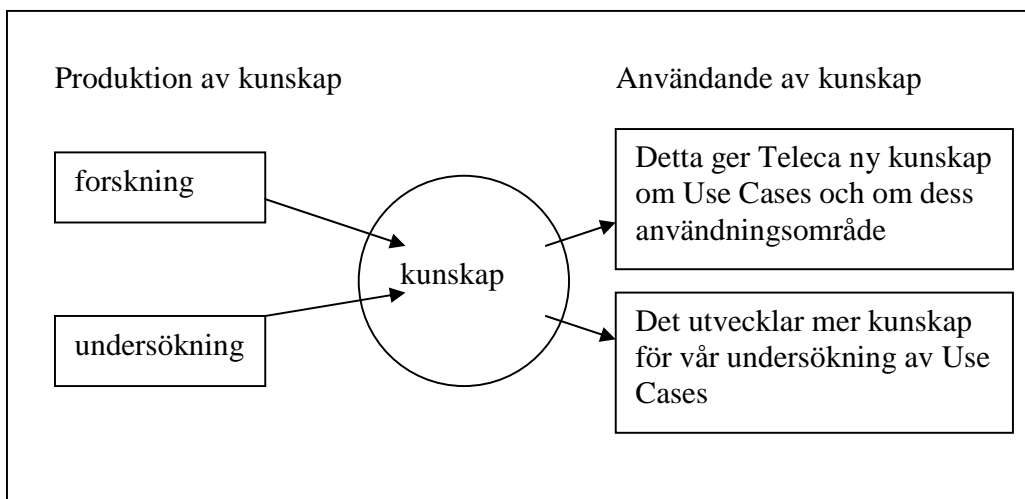
En avgränsning inom projektet som ligger mer på det fysiska planet är Telecas säkerhet. Företaget arbetar inom ett väldigt utsatt område, telekommunikation, vilket gör dem väldigt sårbara för intrång. Därför är säkerheten högprioriterad, vilket gjorde att vi fick signera ett spridningsdokument. Det har gjort att viss information har fått tas bort. Detta har gjort att vår uppsats har blivit mer komplicerad att skriva. Visst material blev sådant vi inte kunde verifiera och det är material som är bortsorterat från uppsatsen. Men vi tycker ändå att vi lyckats kringgå den problematiken. Vi anser att vårt försök att jämföra Telecas Use Cases med litterära Use Cases inte har påverkats.

2 Metod och undersökning

För att förstå hur vår metod och våra instrument har arbetat för oss, vill vi förklara hur kunskapsprocessen har skapats för uppsatsen. Detta för att kunna binda ihop vår litteraturgenomgång med empirin och på så sätt visa hur metoden och dess instrument har varit ett verktyg för oss i vår uppsats.

2.1 Kunskapsprocessen för uppsatsen

Linjen mellan utvecklingen, undersökningen och forskningens delar kan vara ganska oklara (Patel, Davidson 1994). Ett exempel är var gränserna går för vad som är forskning och vad som är undersökning. Detta är beroende av vilken kunskap som en person innehar. Det i sin tur ger en aspekt på skillnader i kunskap. En möjlighet som vi har sett är att skilja på produktion av kunskap och utnyttjande av kunskap, som t.ex. att det skapas nya frågor till kunskapsproduktionen samt att dessa ger mer och ny information om Use Cases (se figur 2-1).



Figur 2-1

Figuren visar hur vår kunskapsprocess har skapats för denna undersökning enligt Patels modell av kunskapsproduktion och användandet av den (Patel, Davidson, 1994 s9).

Vi har försökt sprida ut de olika kunskaperna genom hela uppsatsen för att få en helhet och förståelse om hur Use Cases fungerar samt om hur de kan återanvändas.

För att få strukturen på uppsatsen har vi använt oss av böcker och kompendier som ”Samhällsvetenskapliga metoder” (Bryman, 2002), ”Att skriva rapporter: Om formen och dess betydelse för innehållet” (Pettersson 1997) och ”Rapporter och uppsatser” (Backman, 1998).

2.2 Urvalsförfarande

I detta skede gick vi igenom de metoder som skulle passa oss i vårt arbete. Vi utgick från att en del av vår undersökning var till viss del av en deskriptiv natur där det var en länk mellan olika typer av Use Case-modeller, både med och utan återanvändningsbara Use Cases. Annars utfördes de större delarna i en metodisk kvalitativ natur, där vi frågade företagspersonal samt gjorde observationer med ostrukturerade intervjuer inom det projekt, vi var inbjudna att följa (Patel, Davidson, 1994). Att genomföra undersökningen till stor del på ett metodiskt kvalitativt sätt föll sig naturligt för oss med tanke på det sättet som vi strukturerade vårt arbete och på det sättet vi följde projektet. Anledningen till att det kändes som ett bra val var för att lyckas med utfrågningarna. Vid dessa var det viktigt att få de intervjuade att känna sig lugna och avslappnade vid intervjutillfällena, eftersom de då i lugn och ro korrekt kunde besvara frågorna. Vår lösning på detta var att intervjua personalen på dess normala arbetsplatser. Det var då lättare att få en bra kommunikation med de intervjuade. Personer som blir intervjuade i sina naturliga arbetsmiljöer känner mindre press, vilket i sin tur gör det lättare för dem att komma ihåg allt det de vill säga (Bryman, 2002).

2.3 Den litterära studien

Vår litteraturstudie började med att vi letade efter information på Internet och i böcker relevanta för vårt område för att se vilken mängd av möjlig information som fanns att tillgå. Bland den litteratur vi fann använde vi oss bland annat av Ivar Jacobson's bok *Object-oriented software engineering: A Use Case driven approach* (Jacobson, 1993), vilket blev en milstolpe för oss. Detta eftersom Jacobson är en av grundarna bakom RUP och skaparen bakom Use Cases samt Unified Modelling Language (UML)-standarden. Några av hans efterföljares arbeten om Use Cases och RUP:s uppkomst blev också användbara för oss. Anledningen till att vi valt att koncentrera oss på just Jacobsons modeller och uppbyggnad är att vi ville göra vår jämförelse med Use Cases som är så lika Use Cases beskrivna i RUP som möjligt.

Det finns även mycket litteratur om Use Cases på Internet som var ytterligare ett sätt att skaffa kunskap om hur de används inom dagens IT-samhälle.

Våra kriterier tog vi från den litteratur om Use Cases som vi hittat från dem som skapat Use Cases, tillsammans med den information vi fått från Teleca. Där sorterade vi ut de skillnader och likheter vi kunde finna samt läste oss till vad som var grunden för ett Use

Case enligt Ivar Jacobson. Med all denna information som vi samlat in skapade vi ett internt informationsarkiv.

Detta möjliggjorde att vi kunde kombinera dokumentinformationen vi fått från Teleca med dokumentstudien från litteratur och källor från Internet. Ur den samlade informationen pusslade vi ihop bitarna som gav oss en grund för vår uppsats. Informationen har koncentrerat sig på skillnaderna i Use Cases och på hur Teleca använder Use Cases, men också på hur de litterärt skildras. Med dessa olika källor, informationen från Teleca kontra vår litterära studie, ansåg vi oss ha den grund att stå på som behövdes för att skapa vår uppsats och för att göra de jämförelser som behövdes för att undersöka om ett Use Cases är återanvändbart.

2.4 Observationer

Våra observationer gjordes med ett kvalitativt perspektiv men det gällde att få en struktur på observationerna och för att kunna skapa det så utgick vi från ett ramverk med frågor som vi ville försöka att få besvarade vid varje observationstillfälle. Vi använde en kodningsram som instrument mot vår datainsamling för att objektiviteten skulle bli korrekt, och ej påverkbar av yttre faktorer som t.ex. företagets ideologi, inom detta projekt. Med objektivitet menar vi att det inte skulle finnas några förutfattade meningar om vad utgången av vad varje observation skulle bli (Backman, 1998). Våra observationer innebar även att vi studerade visuellt på Teleca. Exempel på vad vi såg var hur organisationen fungerade eller hur hanteringen av Use Cases fungerade. Detta genom att vi fick tillgång till de Use Cases som skrevs. Under uppsatsens gång fick vi tillfälle att se hur teamwork på Teleca fungerade tack vare att vi kunde röra oss ganska fritt inom Telecas organisation. Vi fick även chansen att pröva på att skapa både vanliga och återanvändningsbara Use Cases. Anledningen var att vi skulle få känna på de skillnader som finns i tankesätt när det skrivs olika typer av Use Cases. En annan observationsform som vi använde oss av var att studera dokumentationen från Teleca. Det bör tilläggas att vi tagit upp viss del av dokumentationen i vår litteraturstudie, men den bör även ses som om att den till viss del ingår i våra observationer. Detta gjordes innan vi riktigt började vår undersökning på Teleca. Dokumentationen gav oss möjlighet att skapa grova strukturer för kategorier till vår kodningsram (se tabell 2.1) med hjälp av litteraturgenomgången som motpol. Detta blev en grund för vårt analysmoment för att underlätta vårt kvalitativa förfarande. Analysen skedde fortlöpande genom uppsatsen (Backman, 1998). Problemet med detta var att sortera ut rätt information. Vi insåg att det skulle bli en större mängd information med olika vinklingar som vid ett tillfälle kanske inte skulle verka intressant men i ett senare skede kunde bli aktuellt för vår undersökning. Därför skapades ett arkiv där vi kunde lägga in information för att senare kunna gå tillbaka för att se på informationen.

Arkivet har varit till stor hjälp för oss i våra genomgångar inför intervjuer och även vid våra diskussioner om vad vi behöver gå in djupare på.

2.4.1 *Frågeställningar med Quick and Dirty*

En av våra datainsamlingars metoder var en som kallas “quick and dirty” (Preece et al., 2002). Denna metod fungerade bland annat som ett verktyg för att kunna validera informationen. Det gjorde det möjligt att bekräfta om det går att använda återanvändningsbara Use Cases inom andra telekomföretag, som extern validitet (Bryman, 2002). Metoden går ut på att använda sig av korta informella möten för att kunna få feedback och svar på ett snabbt sätt. Tyngdpunkten i metoden ligger på att få en snabb respons och inte på att få ett väldokumenterat resultat. Dessutom är det tänkt att metoden ska ta kort tid så att den information som inhämtas snabbt ska komma till användning för att inte glömmas bort. Resultatet blir ofta av en beskrivande och informell art, så som anteckningar, skisser och anekdoter. Resultaten från dessa möten förde vi in i vårt arkiv.

Anledningen till att metoden användes är att vi vid ett flertal tillfällen var tvungna att få respons snabbt från våra kontaktpersoner utan att det skulle ta för mycket tid från dem. De hade ett stort ansvar på sina axlar och därmed ibland begränsat med tid över åt oss och våra frågor. När vi använde metoden var våra frågor oftast av en stängd art. Frågor av stängd art är frågor som kan besvaras med precisa förutbestämda svar (Preece et al., 2002). Användningen av metoden var motiverad eftersom vi i vissa fall behövde validera information vi skaffat tidigare i vår undersökning. De stängda frågorna var en fördel eftersom vi ville ha insamlad data av en kvalitativ natur (Bryman, 2002). Att då använda sig av frågor med förutbestämda svar, ett svar av stängd art, gjorde att vi kunde få ett konkret svar på om det var ett visst alternativ som gällde eller inte. Fördelen med vårt sätt att genomföra metoden, märktes också på, att det vid intervjuer med frågor av stängd art inte fanns något behov av några mätskalor för att nå våra mål. Vikten låg mer i vad den enskilde personen gav oss i verbal form eller i text (Engquist, 2001), vilket vi sedan kunde analysera med vår kodningsram. Det gör även att reliabiliteten ökar, men vissa delar är alltid slumpartade när det genomförs undersökningar i projektform. Det kan vara så att alla de förutsättningar vi haft inte kan återupprepas exakt eftersom projektet är unikt i sig själv. Därför har vi valt att lägga tyngden på tillförlitlighet, överförbarhet och pålitlighet av informationen (Bryman, 2002). Tillförlitligheten genom våra återkommande genomgångar som är nerskrivna i arkivet och som det sedan diskuterades runt. Överförbarhet går att hänvisa till arkivet i sig själv. Pålitlighet genom att våra frågor oftast är dubbelkollationerade från olika informanter (Bryman, 2002).

En nackdel med de få öppna frågorna vi ställde var, att det hände att de inte alltid följde den röda tråd i frågeställningen, som vi tänkt oss från början. Svaren kunde då bli något annorlunda än vad vi väntat oss. Det gjorde, att det ibland kändes som att vi inte hade full kontroll över intervjusituationen (Preece et al., 2002).

Vi försökte att minimera problemet med kontrollen under intervjuerna genom att använda oss av ostrukturerade intervjuer så ofta vi kunde, vilket var beroende på om och när personalen på Teleca hade tid att delta. Vid dessa tillfällen använde vi oss av mer förberedda frågor, där vi använde oss av intervjuscheman vilket ledde till svar som var mer förutsägbara. Möjligheten att ställa de frågor vi ville finns fortfarande kvar i en ostrukturerad intervju och möjligheten att vara flexibel i frågeställningen finns även i de

ostrukturerade intervjuerna (Bryman 2002). För att rangordna svaren från Quick and Dirty och våra ostrukturerade intervjuer så använde vi oss av vår kodningsram (se tabell 3.1). De förberedda frågorna ledde också till att det var lättare att få en röd tråd på de frågor som ställdes under intervjun.

2.4.2 Ostrukturerade intervjuer

De intervjuer vi gjorde bygger på ostrukturerade intervjuer med vissa tendenser åt det semistrukturerade hållet. Hur ostrukturerad en intervju är beror på hur stort inflytande den som intervjuar har på processen (Bryman 2002). Den ostrukturerade intervjun blir mer lik en konversation med fokus på det specifika området. Många gånger kan en ostrukturerad intervju leda djupare in i det ämne som diskuteras. För att säkerställa att det ska uppnås ett önskvärt resultat med en ostrukturerad intervju bör det finnas en mall (Intervjuschema) och en plan att utgå ifrån som innehåller de ämnesområden som ska diskuteras (Bryman, 2002). Ofta leder en ostrukturerad intervju till rikliga mängder av insamlad data. Mängden data kan generera problem då det gäller att kunna sälla ut den information som är viktig (Preece et al., 2002)

Vår mall för våra intervjuer skapades innan vi höll mötet med den intervjuade, vilket ofta var innan de informationsmöten som vi hade tillsammans med deras personal. Vid de tillfällena gick vi igenom vilken eller vilka frågor vi ville ha besvarade och då var inte den intervjuade med. Vad vi ville ha ut ur intervjun var inte alltid det som kom ut ur den. Mötena hölls för att frågan/frågorna skulle vara byggda på samma konceptuella modell, så att det inte skulle råda några oklarheter om vad det egentligen handlade om (Bryman 2002). Detta arbetssätt gav oss ett urval av frågor att välja från inför varje intervju. De ostrukturerade intervjuerna gav oss både öppna och stängda svar. Informationen vidgade hela vår syn på uppsatsen både positivt och negativt. Positivt i den bemärkelsen att personalen var lätt att föra diskussioner med. Det negativa var att hålla intervjuerna i fokus på vårt mål. Andra fördelar med att arbeta på det sätt vi valt är att det var lätt att följa upp de svar vi fick och sedan hitta andra vinklingar på frågeställningarna. Det var lätt att ställa uppföljningsfrågor under intervjuns gång och det gick också enkelt att sticka in extra sonderingsfrågor för att få en fördjupning av de svar som gavs på våra frågor. Detta kunde leda till att vissa vinklingar, som kunde te sig som ointressanta sidospår, senare visade sig vara riktigt intressant för uppsatsen (Bryman 2002).

Nackdelen med dessa ostrukturerade intervjuer var att det var väldigt svårt att sortera ut den information som var relevant för uppsatsen, eftersom den insamlade informationsmängden ofta blev väldigt stor. Vid varje tillfälle så hade vi med oss våra frågor nedskrivna på ett papper på vilket vi under intervjuns gång kunde notera de svar som gavs. Vi hade även tillgång till diktafon där det var möjligt att spela in intervjun på band. Dessa inspelningar kunde sen spelas upp för att återge intervjun och på det sättet säkerställa informationen (Bryman 2002).

2.4.3 Intervjuschemat, visuella ramverket och kodningsramens utseende

Vi byggde mycket av våra öppna frågor på det område vi ville ha svar på och med ett språk som var tydligt för den som intervjuades. Vi undvek ledande frågor och försökte hålla det så könsneutralt som möjligt, däremot tog vi upp vilka positioner de intervjuade hade (Bryman 2002). Intervjuerna utgick till stor del från följande grundfrågeformulär vilket har gett en mycket stor del information som vi kunnat analysera. Ett intervjuschema såg exempelvis ut så här.

1. Vilket ramverk används, vilket ramverk har de utgått från och varför?
2. Vilken typ av Use Case finns det på företaget, använder de dem som Grafisk notation(UML)/User scenario stil?
3. Hur används Use Cases rent praktiskt på företaget, hur definierar de ett Use Case?(Vilka beståndsdelar finns det, är någon del viktigare än en annan, hur arbetar de med dem inom projekt)
4. Återanvänds eller vidareutvecklas (ommodelleras) befintliga Use Case i dagsläget?
5. Vilka Vinster/möjligheter tror/vet de att de skulle få/har med hjälp av återanvändning?
6. Finns det någon spårbarhet (traceability) från Use Cases ut i övrig verksamhet och dokumentation? (Supplementary Specifications, Glossary, Business Case, Vision)
7. Hur arkiveras Use Case på företaget?
8. Har återanvändning påverkan på den slutliga produkten?
9. Hur namnges ni Use Case på företaget?

Eventuella följdfrågor som kom upp fick anpassas efterhand som projektet gick vidare in i andra faser. Förändringen berodde på vilken information som vi saknade för analys till det kvalitativa perspektivet (Backman, 1998). När det gäller kodningsramen, där vi satt upp våra kategorier, så användes den både för intervjuer och för att både verifiera och säkerställa information från hela projektet. Informationen som vi införskaffat från undersökningen analyserade vi med materialet från den litterära informationen, vilket resulterade i data som sedan kunde användas i en diskussionsdel. En annan del för evalueringen i projektet var att vi visuellt observerade det som hände runt omkring oss. För att få någon struktur på detta skapade vi ett visuellt ramverk (Bryman, 2002). Exempel på frågor som vi ställde oss för vårt visuella ramverk såg ut som följande:

- Vem/vilka använder sig av tekniken/momentet som studeras?
- Var befinner den/de sig?

- Vilka rutiner använder de sig av?
- Hur betar de sig?
- Vad gör de med tekniken/momentet?
- Finns det regler och normer som påverkar arbetet?

Anledningen till att vi valde de här frågorna var för att se om resultatet av arbetet påverkades av vem som utförde det. Samtidigt fick vi följa flödet på Use Cases och se hur många som påverkades av det, som t.ex. de som skriver manualer. Hur reglerna på Teleca påverkade Use Cases genom hur de var tvungna att skriva dem så att de skulle fungera. Vidare om platsen de utförde det på hade någon betydelse. Om en viss rutin eller ett visst beteende påverkade resultatet (Bryman, 2002). Vi ville också se hur de utnyttjade Use Cases och om det fanns några förutbestämda normer och regler som gjorde att resultatet av arbetet blev som det blev. Det är inte givet att det måste vara rätt, för att det används ute i den reella världen men det ger en antydning om att det fungerar.

Detta gav ett fysiskt verktyg att arbeta med, som spelade en viktig roll. Tilläggas ska också att våra visuella observationer i grunden var ostrukturerade. Därför skapade vi det visuella ramverket (Bryman, 2002). Organisationen som höll i projektet försåg oss med sina verktyg, projektrum och datorer med Internetuppkopplingar. Med sådan uppbackning är det svårt att inte vara partisk, men för att kunna följa projektet på så nära håll, blev det ett måste att använda deras utrustning. Samtidigt bör det poängteras, att de ville ha en objektiv syn på våra resultat.

En viktig fråga är ”vidareutveckling av ett system” vilken innebär att det ändras i vissa delar inom systemet och även hur Use Cases då används. Räknas då de Use Cases som används igen som återanvändningsbara Use Cases? Alla de här differenserna passar att lägga in i den kodningsram som vi skapade. Kodningsramen är i grunden de kategorier (se tabell 2.1) vi fann under vår start av observationerna samt Telecas dokumentation och i den litteratur som vi gått igenom.

Till kodningsramen (se tabell 2.1) använde vi oss sedan av följande kodning på frågorna och dess svar, dvs. våra frågor ställda med Quick and Dirty metoden eller för frågor ställda under våra ostrukturerade intervjuer, för att avgöra vilken information som var viktigast. Även annan information som vi hittade i dokument och litteratur och fick beskrivna som viktig, exempelvis vad författarna lagt mest tyngd på, eller där någon information särskilt poängterades fördes in i kodningsramen som om den hade extra viktig betydelse.

Tabell 2-1
Kodningsram

Påverkas	intervju	visuellt	Teleca dokument	Litteratur
Spårbarheten				
Funktionalitet				
Användbarhet				
Pålitlighet				
Prestanda				
Support				
Arkivering				
Namnkonvention				
Use Case dokumentationen				
Gränssnittet i slutlig produkt				

1. Oviktig information
2. Viktig information
3. Mycket viktig information

Sedan graderade vi denna information genom att fylla i kodningsramen med 1, 2 eller 3 och därigenom fick vi fram det som var viktiga punkter för vårt resultat. När det gäller intervjuer så var det högst antal poäng, som styrde vikten. Det gjorde också, att vi kunde sälla vår information och ta fram det, som hade värde för projektet, och samtidigt få ökad reliabilitet med våra dubbelkontroller mellan olika litteraturer, så att vi kände, att vårt material hade lett fram till det, som hade styrt våra beslut (Bryman, 2002).

2.5 Våra beslut och val

Vi har tagit vissa beslut för våra metoder för att kunna genomföra våra undersökningar och de besluten har varit ett skäl till hur strukturen ser ut. Vi började se över det som kan kallas ett standard-Use Cases enligt RUP, sen tog vi Use Cases som Teleca anser vara återanvändningsbara Use Cases och jämförde likheter och olikheter mellan dem. Detta bildade en stomme med hjälp av de instrument vi tagit upp tidigare, för att se om likheterna var så pass stora att det går att kalla det för ett återanvändningsbart Use Case inom ramverket RUP. Om det inte går att undvika förändringar i ramverket så är det inte samma teknik utan en helt ny teknik. För att få den informationen som behövdes passade vår observationsmetod in.

Våra instrument som Quick and Dirty användes för att kunna få snabba svar och verifiering av redan ställda frågor som vi behövde ställa till personalen på Teleca.

Metoden var en fördel eftersom vi ibland behövde ställa några korta komplementerande frågor för att säkerställa att vi hade uppfattat allt korrekt.

Metoden var också användbar för att kunna ställa några inledande frågor för att få en första idé om ett område, vi inte täckt in tidigare, för att se om det var något som behövde analyseras djupare.

Vi insåg att mängden information redan i litteraturstadiet skulle bli överväldigande.

Därför byggde vi upp ett internt arkiv för vårt projekt, och till det formulerade vi ett stort antal frågor baserade på litteraturen i vårt arkiv. Arkivet bestod av anteckningar om var den relevanta informationen fanns att tillgå och på vilket sätt den var relevant för området vi skulle observera. Dessa frågor blev sedan en stomme för både vårt intervju-schema och det visuella ramverk, som vi placerade i vår kodningsram, där vi rangordnade de frågor som vi ställde. Våra grundfrågor blev genast till nya som vi också fick ta hänsyn till. Vi diskuterade alltid alla frågor innan vi skulle intervjua. Efter intervjuerna använde vi oss av kodningsramen igen för att rangordna och gå igenom svaren för att sortera ut den information som var viktig.

För att leda oss till vilka personer vi behövde fråga ut, gick vi före intervjuerna igenom, vem som hade tillgång till de önskvärda informationerna. Här styrdes vi till stor del av vilken tillgång och vilken tid, vi kunde få med personalen på Teleca. Det hände att vi fick vänta in personer, eller att projektet var i ett skede där vi enbart fick stå vid sidan om.

Dessutom hade projektet en tidsram att följa, vilket ytterligare satte begränsningar. Vi fick ändå betydligt mer tid och tillgång till deras personal, än vad vi hade trott från början, och vi har inte funnit någon anledning, att misstänka att den något begränsade tillgången till Tecas personal har påverkat slutresultatet negativt i undersökningen.

Vi har fått tillgång till de nyckelpersoner som vi velat träffa, och vi anser oss också ha fått tillfälle att ställa de frågor som vi önskat, enligt det intervju-schemat vi satt upp.

Tack vare att vi fick följa med i ett projekt fick vi möjligheten att visuellt observera hur det gick till i ett projekt. Samtidigt gav det oss en anledning att välja att visuellt observera. Anledningen till att visuellt observera var att vi ville få en objektiv bild av hur vissa moment gick till, och för att se hur personalen påverkades av att arbeta med återanvändning av Use Cases. Det var inte alltid så lätt för de anställda att förklara vad de gjorde och varför de gjorde det på ett visst sätt.

Vid vår visuella observation fick vi även inblick i hur de använder sig av den återanvändbarhet som vi undersöker i vår uppsats. Detta kan på sikt visa vilket resultat och även vikt av det arbetet vi utfört, samt hur effektiv metoden är och ”om” och ”hur” personalen påverkas både fysiskt och psykiskt.

Det kan vara svårt att skilja på ett ommodellerat Use Case och ett återanvändningsbart Use Case. I början av vår undersökning hade vi tankarna att ommodellerbara Use Cases var återanvändningsbara. Därför förde vi in dem som återanvändningsbara och att det var projektet som styrde om de var det eller inte. Exempelvis ansåg vi att ett nytt projekt med gamla Use Cases är lika med återanvändningsbara medan gamla projekt med gamla Use Cases var ommodellering.

Den information vi har samlat in från det här projektet visar att vissa förändringar måste ske för att det inte ska bli några tomma luckor mellan Use Cases som blir involverade i nya projekt.

När det gäller säkerheten på Teleca så har den ställt till med vissa problem för oss. Teleca hade ett par krav när vi gjorde våra intervjuer. De ville inte att intervjuerna skulle publiceras eftersom delar av deras marknadsstrategi då kunde avslöjas. De ville inte att vi avslöjade huvudmålet med projektet som vi fick observera eftersom det skulle kunna avslöja inom vilket område Teleca vill arbeta i framtiden. Vi har valt att tillmötesgå Telecas krav eftersom vi annars inte hade fått göra några intervjuer. Det är också anledningen till varför vi inte har tagit med transkriberade intervjuer i uppsatsen. Vi tror inte att reliabiliteten på undersökningen har påverkats av detta. För en utomstående så blir det förstås svårt att verifiera om reliabiliteten har påverkats. Vi har dock haft det i åtanke och försökt att verifiera de svar som vi fått från så många källor som möjligt, så att den interna reliabiliteten på undersökningen ska kunna säkerställas, även om det är svårt att verifiera för en utomstående i efterhand. Ett ytterligare krav var att en del av deras dokumentation som vi hade tillgång till inte fick publiceras utanför Teleca, till exempel "Use Case Report" och projektens dokumentation. Vi är medvetna om att avsaknaden av denna information i uppsatsen kommer att göra det väldigt svårt att replikera vår undersökning, men vi kunde inte hitta något sätt att komma runt detta problem.

3 Litteraturgenomgång

I detta kapitel kommer vi att ta upp urvalet av litteratur som vi valt för vår undersökning, hur det gått till och varför vi anser att de varit de bästa. Vi kommer även att gå igenom hur vi har använt och arbetat med den. Vi har tagit med fakta som framkommit av våra litteraturstudier.

3.1 RUP:s och Use Cases historia

Utvecklingen startade i sent 60-tal och skälet var bland annat att kunna hantera krav, via Use Cases, på ett enklare sätt, hantera och skapa kravspecifikationslistor från kunder och att analysera modellens funktionalitet. Efter trettio års utveckling så var RUP redo att släppas till allmänheten 1998. Vi vill bara poängtera att det är viktigt att skilja på att RUP är hela ramverket och att Use Cases är en modell inom RUP, så att de inte blandas ihop. Slutprodukten var ett resultat av ett antal teams arbete, men de som höll i trådarna var speciellt tre stycken mjukvaruutvecklare som dubbade sig själva till "the Three Amigos", Grady Booch, James Rumbaugh och Ivar Jacobson *Face to face* (Öhman, 2006). De är mest kända för utvecklingen av UML och deras bok *The Unified Software Development Process*. Boochs huvudansvar var utvecklingen av Rational Rose, vilket gör att användaren enkelt kan skapa en komponentbaserad arkitektur. Rumbaugh är en av de ledande mjukvaruutvecklarna av metodologier i världen, speciellt inom området realtidsmetodologi. Den tredje medlemmen av gruppen, Ivar Jacobson, är en av dagens mest omtalade mjukvaruingenjörer, dels för hans arbete på Ericsson där han var med och startade arbetet på RUP, skapare av Use Cases samt stora engagemang i UML utvecklingen (Cockburn, 2001).

RUP är en fortsättning av succén Rational Objectory Process (ROP), som Jacobson arbetade med under sin tid på Ericsson. Där använde de något som heter "traffic cases." Dessa var liknande Use Case-diagram, då de användes för att representera olika typer av telefonsamtal. Jacobson såg att det var ett kraftigt designverktyg och insåg att en sådan vinkling kunde bli användbar för andra applikationer. Han inkluderade alla aspekter på mjukvaruutvecklingen inom livscykelmodellen och då närmade han sig RUP. Redan på den tiden fanns inom mjukvaruarkitekturens område många objektorienterade modellspråk, och det var även på den tiden som UML skapades av "the Three Amigos" (Eriksson, 2004). RUP definierade hela modellanvändandet i UML. UML används bland annat som ett verktyg för att modellera Use Case-diagram, en term given av Jacobson 1986 (Scott, 2001).

Jacobson fick frågan i en intervju 1999 om han förväntade sig att RUP skulle ha en sådan inverkan och bli en lika stor succé som UML hade haft på mjukvaruarkitekturen; han svarade.

Yes, absolutely yes, and we have very good reasons to believe that. We are making inroads into many corporations today, and it's our goal to get there. We don't think it would be an easy thing to make the Unified Process a standard, it would be so much hard work and so much opposition, so we'd rather do it in small steps. Instead of going and forcing people through a standard, let people convince themselves. And I think that everyone that looks at the Rational Unified Process will become convinced this is the way they've got to do it, as soon as they have started to look at it. There is nothing even close to it. Many people tried to say that there is, but what is that they have? They have something that can be compared with my book, but they don't have anything that can be compared with our process. If you just look upon it in terms of substance, and depth, and experience, and so on, and if you compare ... How old is Approach A, or Approach B? Do we know that it works, for large projects? We know that our works. It's really very different.
(Comani, 2006)

3.1.1 Vad är RUP

RUP är ett hjälpverktyg och ett ramverk som används mestadels inom systemutveckling. Det är uppbyggt på många olika modeller för att undvika misstag inom utveckling av system. När RUP introducerades var det ett revolutionerande verktyg för modellerarna. Där fanns olika modeller för allt som ett projekt kan behöva. Det uppdaterades via nätet som vilken applikation som helst, vilket var en baktanke som författarna haft, för att förhindra att den skulle hamna i en bokhylla och bli bortglömd. Ramverket gav minimalt med oönskade problem som inte fanns med i beräkningarna. Problemen kunde förutses innan de dök upp, vilket medförde att det gick att hantera dem innan de blev ett problem. Att använda RUP har blivit ett av de mest populära sätten att bygga objektorienterade mjukvaror. Den beskriver på vilket sätt krav samlas in, hur de analyseras och hur det går att bygga och designa mjukvaran samt testa den på ett iterativt sätt (Kruchten, 2003).

3.2 Vad är Use Cases

Året 1986 introducerade Ivar Jacobsen en ide, om hur funktionella krav kan beskrivas, som kallas Use Cases (Jacobson et al., 1993). I en artikel skriven av Ivar Jacobsen förklarar han att han påbörjade sin forskning om Use Cases 1967 men att hans forskning inte fick något erkännande förrän i mitten av 80'talet.

It was 1986; I'd worked at Ericsson for more than twenty years and was trying to figure out a better way to model telephone calls. The modern switches of that time handled so many different types of telephone calls: local

calls; outgoing calls; incoming calls; transit calls. There were also many kinds of local calls and many kinds of outgoing calls: calls to a neighbour switch; calls to a domestic switch; calls to an international switch. And, on top of this, each one of these calls could be carried out with different signalling systems. We had been grappling with the problem of modelling multiplicity and diversity for many years. It would have been very difficult to model each type of call - there were too many, and there was a lot of overlap between them. So first we listed and named them. We called them "traffic cases." Our approach was to model the different "functions" or "features" needed to carry out all the calls - all the traffic cases. Functions had no interfaces. They had beginnings and endings, but they were not well defined. A function could interact with the outside world. The general feeling was that we didn't really know what functions were, but we could give examples of them, and some people could specify them. However, we did know how to realize functions. I had learned a diagramming technique to describe sequences of relay operations; in 1969 I translated this technique to describe software component interactions. I called the resulting diagrams "sequence diagrams," and we still use that name for them today. With sequence diagrams (or collaboration diagrams for simpler interactions) we described how functions were realized, in very much the same way we describe use-case realizations today. (Jacobson, 2006)

Under starten på vår undersökning insåg vi nog inte hur och i vilken omfattning som Use Cases används, men vi upptäckte vilket kraftfullt och erkänt verktyg det är för att specificera funktionella mjukvarukrav med. Det finns dock fortfarande återkommande problem som dagens modellerare stöter på. Ett allvarligt problem är hur de ska specificera de alternativa vägar som ingår i ett Use Case.

Från tidigare erfarenheter så kan modellerarna se att de måste ta med de alternativa vägar som finns inom ett visst affärsförhållande. De måste också ta med de delvis parallella och de helt parallella vägar som finns i ett Use Case, samt de eventuella undantagsfall som kan finnas i ett Use Case (Cockburn, 2001).

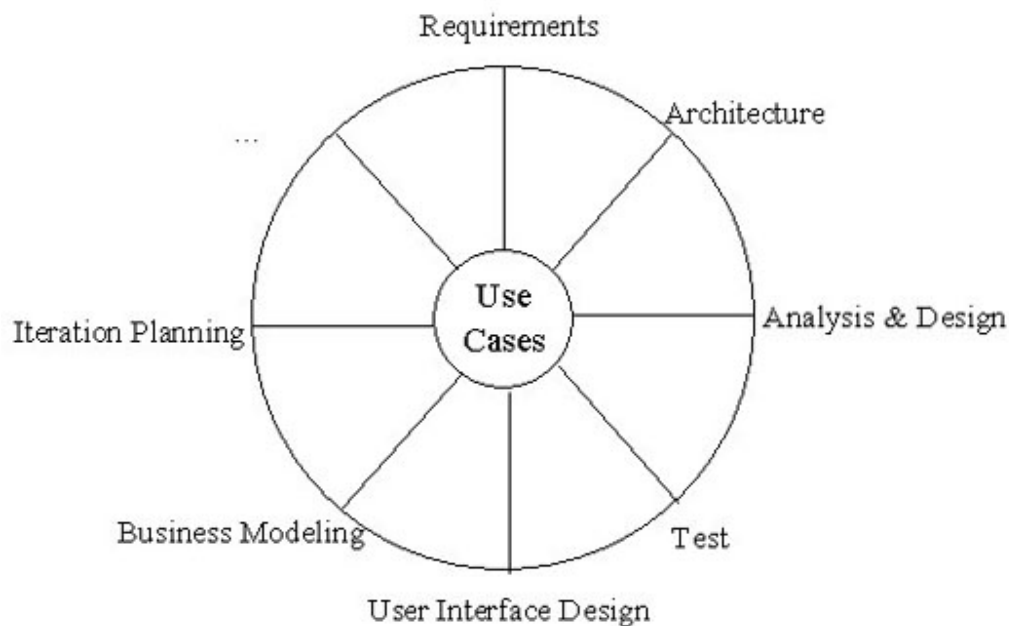
Idag finns Use Cases inom andra ramverk som exempelvis Extreme Programming, XP, som bygger på något som är liknande Use Cases med en lättare form för att ta fram krav för beteende. Här är det vanligt att affärssidan sitter vid sidan om utvecklaren och skriver användarhistorier istället för scenarion (Cockburn, 2001).

Detta visar vilka användbara verktyg de är, men gemensamt för dem är att Use Cases hjälper modellerarna att upptäcka vilka krav och behov som är avgörande i de projekt som de är involverade i. Att använda sig av Use Cases kan reducera tidsåtgången i projekt och det ger modellerarna en god bas som de kan analysera för att säkerställa att de uppnår det önskade resultatet. Men som vi ser på saken så hade ytterligare tid kunnat sparas om det hade gått att återanvända dem. I dagsläget så används Use Cases inte enbart inom systemutveckling. De används inom många olika områden, som till exempel för att visa flödet inom en industri (Guiney et al., 2003).

Redan i början på nittioalet hade Use Cases formats så att de ser ut som de gör idag. Målet var att ge utvecklarna ett verktyg som kan utveckla visioner och kreationer till verkliga system gjorda med de krav som användarna vill ha (Bittner, Spence, 2003).

För att förtydliga skillnaden mellan instansen (äldre Use Cases) av Use Cases mot beskrivningen av Use Cases var att beskrivningen innehöll följande delar, en kort redogörelse för syftet med Use Cases, för att sedan övergå till att beskriva kontrollen över flödet i dem. Huvudflödet och alternativa flöden som i viss mån kan ha delflöden som skulle kunna vara återanvändningsbara inom samma Use Case beskrivning, detta eftersom händelsen räknas som unik. På Teleca ser de delflödena i Use Cases som viktigare än vad de gjort tidigare. Detta eftersom det är dessa som troligen kan gå att återanvända.

Use Cases är mer än en kravframtagande teknik, den är som ett nav i ett hjul.



Figur 3-1

Use Cases är mer än ett sätt att ta fram krav de kan ses som ett nav på ett hjul. De blir som spindeln i nätet. Det visar att många kan arbeta inom flera områden med samma Use Case på samma gång.

Use Cases: Yesterday, Today, and Tomorrow (Jacobson, 2006)

Detta gör att det går att arbeta med samma Use Case på flera plan samtidigt, vilket medför att ett projekt kan genomföras på ett mycket snabbare sätt. Exempelvis kan manualer börja skrivas redan vid utförandet av programmeringen, och beslut kan skötas på ett effektivare sätt. Detta ger fördelar som spårbarhet för analys, design, implementation samt att test både påskyndas och säkerhetsställs. Detta stärker ramverkets position som ett verktyg för framtagande av krav. Vad själva Use Cases olika delar består av tar vi upp i ett senare skede.

3.2.1 *Möjligheterna till återanvändning av Use Cases idag*

Det finns en teori om att kunna återanvända Use Cases som bygger på vad man kallar Generalizations Use Case vilka är abstrakta. Det är när man har ett huvud-Use Case (förälder) och i dem har Sub-Use Cases (barn).

Sub-Use Cases kan användas så fort där finns en instans av föräldern i det, det är här teorin bygger på återanvändning. Av instansen krävs det att regler finns för att barnen ska kunna kräva andra interaktioner mot en aktör. Exempelvis ses, att ”ringa ett lokalt samtal” (barn) eller att ”ringa en telefonservice”, som generaliserade till ett huvud-Use Case ”ringa upp” (förälder). Därvid tillförs det ett beteende till huvud-Use Cases (förälder) utan att det behöver ändras i det (Jakobson, 2003). På detta sätt skapas återanvändningsbara Use Cases.

Att använda t.ex. extension är en teknik som är lätt att förstå. Extension tekniken bygger på att uttrycka varianter (extension) vilka kan användas som bifogade delar till extension points, det används sen i huvud-Use Cases för att koppla Sub-Use Cases. Det börjar med att beskriva grunden från basflödet för att sen se om det finns något extra beteende som behöver förklaras. Inom Use Cases är flöden huvudpunkten om ett Use Case ska vara möjligt att återanvända.

För att verkligen kunna återanvända Use Cases behövs det en djupare kunskap om hur Use Cases fungerar. Annars blir det ohållbart att hantera dem, eftersom tekniken är av en annan typ (Jakobson, 2003).

När det gäller namnkonventioner så finns dessa även inom programmering av IT-system. Detta belyser vikten av att använda standardiserade namnkonventioner även i Use Cases. Det är viktigt eftersom programmeringsspråk som Java bygger på återanvändbara bibliotek där man kan utnyttja funktionalitet i återanvändningsbara syften (Skansholm, 2002). Det är på samma sätt som Ivar Jacobsson bygger upp sina teorier om att kunna skapa bibliotek (arkiv) som bygger på Use Cases egna funktionaliteter (Jacobsson, 1997).

3.2.2 *Nästa generations Use Cases*

Rational anser att framtidens Use Cases och återanvändbarhet är en viktig del av utvecklingen för framtida Use Cases. De teorier som finns om helt återanvändningsbara Use Cases är i forskningsstadium och är under utveckling. Det innebär att inget är fastslaget om hur de kommer att se ut i framtiden. Det behövs en djupare förståelse för Use Cases, både affärsCases samt de som används för mjukvaruutveckling och som det fokuseras på i boken *Software Reuse* som Jacobsen publicerade med Martin Griss och Patrik Jonsson på slutet av 1990-talet. “*Software Reuse: Architecture, Process and Organization for Business Success*” (Griss, Jonsson 1997). Vi har uppfatta det som att det idag är mer relevant än någonsin med återanvändningsbara Use Cases när det byggs integrerade IT-system som behöver support. Till viss del kan även vidareutveckling av system bygga på Use Cases som kan återanvändas utan att ommodelleras.

Huruvida återanvändning är en av de framtida finesser som finns med Use Cases är något som återstår att se.

Möjligheten att återanvända Use Cases är och kommer att bli en stor förändring vid hantering av nya applikationer, men det gäller dock att observera skillnaderna på ett ommodellerat Use Case och ett helt återanvändningsbart Use Case. Idag skrivs många applikationer om och förnyas och då används gärna de tidigare använda Use Cases igen med den skillnaden att de modelleras om.

Att skriva om applikationer eller uppgradera dem bör startas med att förstå Use Cases (Jacobson et al., 1997).

3.2.3 Användningsområden

Ett Use Case förändras beroende på vad en modellerare vill göra, och beroende på vad modelleraren vill använda det till. Till exempel så används idag denna typ av modellering inom flera områden i industrin och systemutveckling. Detta komplicerar den formella definitionen av vad som är ett Use Case, men Ivar Jacobson definierar det så här enligt Kruchten.

Use Cases provide a means of expressing the problem in a way that is understandable to a wide range of stakeholders: users, developers, and customers. (Kruchten, 2003, s.98)

Detta öppnar naturligtvis upp för en mängd olika tolkningar av vad som är ett Use Case och hur det ska användas. Vad vi vill visa med detta är att det är fullt möjligt att använda sig av Use Cases trots att det inte finns en objektorienterad analys som bas för dem, även om så ofta är fallet. Vi vill uppmärksamma att detta inte är ett fel i Use Cases. Use Cases är ett analysverktyg som kan appliceras på ett brett spektrum av användningsområden, så väl objektorienterade som icke objektorienterade. Detta gör bara Use Cases ännu mer användbara.

3.3 Viktiga begrepp i Use Cases

Vi har redogjort för hur Use Cases utvecklades och hur framtiden ser ut för denna teknik, samt vilka användningsområden den har. Nu ska vi gå djupare in i dem och granska de begrepp som Use Cases innefattar och skälen till dessa begrepp. För att kunna arbeta med Use Cases så är det viktigt att förstå, vilken stor betydelse det har att kunna spåra och följa en röd tråd genom ett projekt. Use Cases används normalt för att hitta de krav eller den funktionalitet, som användaren kan tänkas ha med systemet (Kruchten, 2003). Detta är begrepp vi kommer att ta upp senare. Anledningen till att Use Cases används på detta stadium är att det är viktigt att hitta alla funktioner som ska ingå i systemet, innan det börjar byggas prototyper. För att en modellerare ska kunna tillverka Use Cases så måste han ha goda kunskaper om notationen i Use Cases så väl som en god förståelse för vad

begreppet aktörer (Kruchten, 2003) innebär. Anledningen till att Use Cases används är för att skapa en grafisk modell över funktionaliteten mellan ägaren (intressenten), användarna och systemet. Från denna grafiska bild så kan modelleraren få en start och en slutpunkt på ett Use Case (se bilaga [UC_IRC_2] Connect Server).

Efter det kan modelleraren skapa ett funktionsflöde som både innehåller ett basflöde och alternativa flöden. Det kan också ha anknytningar, till ett annat Use Case, vilka är en delmängd av ett Use Case. En sådan anknytning kallas en anknytnings punkt (extension point) vilken specificerar en unik punkt i basflödet. Modelleraren bör kontrollera om det finns anknytningspunkter och om det finns några relationer eller icke funktionella krav (Kruchten, 2003) som tillhör ett Use Case (se bilaga Use Case image Teleca). När modelleraren har kommit dit har han en modell av vad systemet har för funktionalitet. Det är då dags att testa Use Cases för att upptäcka om någon eller några funktionaliteter har missats (Cockburn, 2001). Det som ska testas är om alla krav som ägaren vill ha implementerade täcks in med de Use Cases som har tillverkats.

3.3.1 Grundkrav inom Use Cases

Ett av grundbegreppen inom Use Cases är krav (requirements). För ett av de främsta syftena med Use Cases är att uppfylla eller att inte missa några krav. Krav är ett tillstånd eller en egenskap som ett system måste uppfylla eller inneha. Nödvändiga attribut i ett mjukvarusystem är kvalitets attribut som på engelska kan förkortas FURPS. Akronymen står för **functionality, usability, reliability, performance** och **supportability** vilket blir ungefär **funktionalitet, användbarhet, pålitlighet, prestanda** och **support**. Dessa är begrepp som ligger inom kravområdet för Use Cases. Denna akronym används för att det ska gå att definiera ett komplett kvalitetssystem (Kruchten, 2003). Krav består oftast av bara en dimension i avseendet, att de oftast består av text som beskriver kravet (Jacobson et al., 1993). När en modellerare har alla krav, eller åtminstone tillräckligt många för att skapa en kravspecifikation för det som krävs av systemet, så kommer de att representera en första modell av den förändring som systemet kommer att genomgå under modelleringen.

Funktionalitet: Funktionalitet är de operationer som systemet utför åt användaren och det är det minimum som ett system måste kunna utföra. Inom funktionalitet finns två begrepp, funktionella- och icke funktionella krav. Begreppet funktionella krav är de som används för att beskriva systemets beteende. De in- och utdata som ett system kräver och producerar ska vara i en form som kan förutses av användaren (Kruchten, 2003). För att uppnå önskad kvalitet måste modelleraren hålla reda på en mängd attribut som inte finns med i systemets funktionella krav (Kruchten, 2003). Dessa attribut är de icke funktionella kraven, som kommer att påverka den design modell, som tas fram. Några av de icke funktionella kraven är användbarhet, pålitlighet, prestanda och support. Icke funktionella krav är ofta krav som ett system måste inneha för att följa lagar och regler som finns utanför systemet. De kan också vara krav som en vald design ställer på systemet, t.ex. att systemet måste fungera med ett annat befintligt system.

Användbarhet: Detta är det krav som hanterar mänskliga faktorer. Användbarhet handlar om hur enkelt ett system är att lära sig. Det handlar också om hur konsekvent ett system är i såväl användargränssnitt som dokumentation och eventuella handböcker och tränings material (Kruchten, 2003). Genom att använda Use Cases blir förklaringen av funktionaliteten tydligare för en kund, så att kunden lättare förstår den funktionalitet som implementeras i dem. När kunden förstått vilken funktionalitet det är som implementeras får systemet ökad användbarhet. Det kan också underlätta när det vid andra tillfällen ska tillverkas manualer för projekt där Use Cases kan vara inblandad. Det är också viktigt att tänka på, att det är i detta skede, som det bestäms hur Use Cases ska kunna användas i framtiden på företaget.

Pålitlighet: Pålitlighet eller hur konsekvent en teknik är kan förklaras med hur väl den kan producera samma resultat vid olika tillfällen och under olika förhållanden (Preece et al., 2002). Precision och förmågan till återhämtning är två av de viktigaste kraven när det gäller pålitlighet. Andra viktiga delar är förutsägbarhet och hur ofta och allvarligt det blir fel i ett system. När det gäller Återhämtning handlar det om hur väl ett system återställer sig efter ett systemfel.

Prestanda: Ett Use Case ställer andra krav utöver de rent funktionella. Ett prestandakrav är, att systemet måste stödja en viss funktionalitet, när användaren kräver det (Kruchten, 2003). Om det ska gå att återanvända Use Cases med tanke på prestanda, så måste samma design användas som använts tidigare, om de ska få samma prestanda som förra gången.

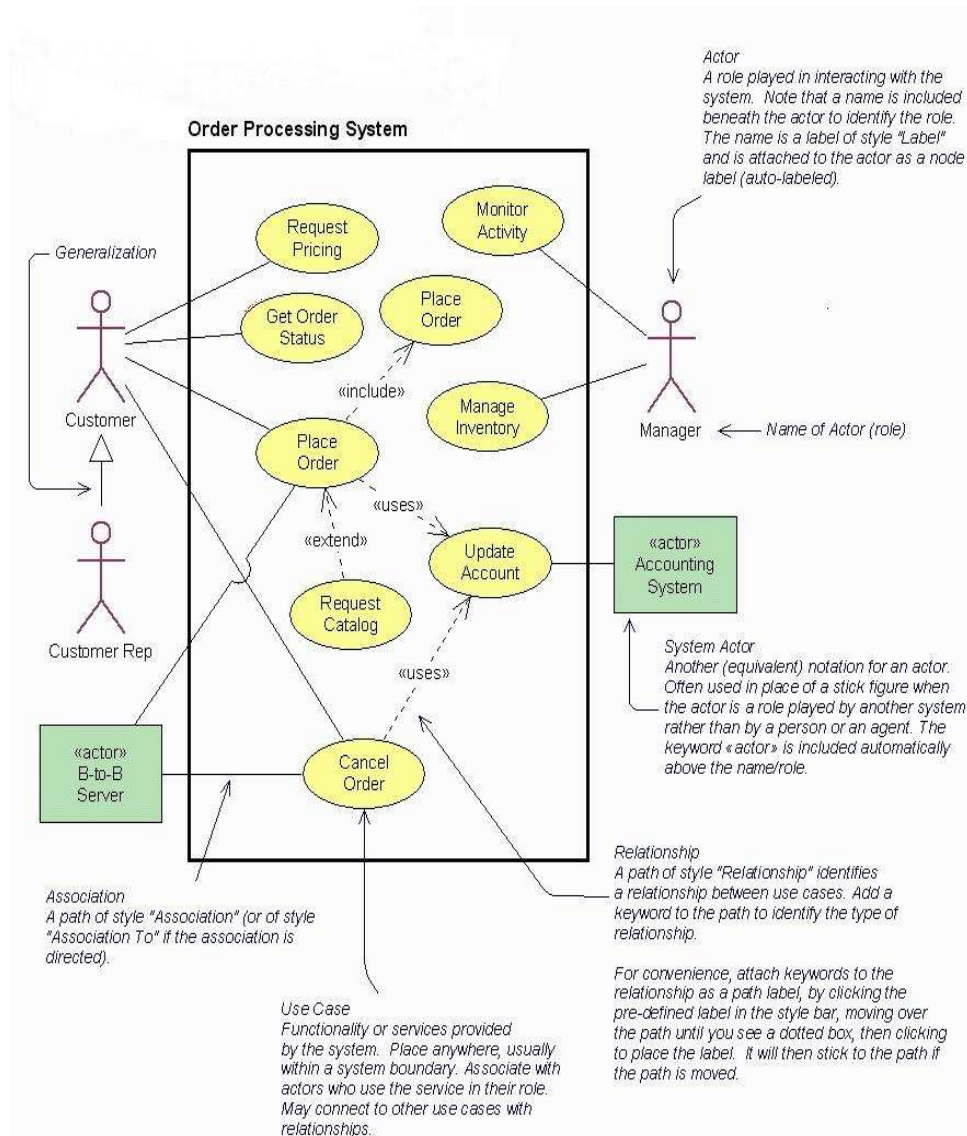
Support: Med support menas vad systemet stöder eller har support för. Detta är ett krav som är unikt på ett sätt, eftersom det inte är ett krav direkt på systemet utan mer ett krav som ställs på den systemutvecklingsprocess som används när system utvecklas och de delar som ingår i systemet (Kruchten, 2003).

Viktiga delar som behöver beaktas vid tillverkning av system är vilken form av **Återhämtning** som systemet ska stödja. Återhämtning handlar om hur väl ett system återställer sig efter ett systemfel. System ska kunna återhämta sig utan att det förlorar data eller att system fel finns kvar när det har återhämtat sig. **Förutsägbarhet** är hur ett system ska bete sig på ett för användare förutsägbart sätt och utföra det som en användare förväntar sig att det ska göra. Detta kan uppnås genom att använda vedertagna symboler och funktioner. **Säkerhet** i ett system handlar om delar, som det är önskvärt att det ska inneha för att förhindra användare från att göra allvarliga eller farliga fel, fysiskt farliga, inkorrekta eller oönskade. (Preece et al., 2002).

3.3.2 Use Cases uppbyggnadsdelar

Även delar inom Use Cases blir begrepp som måste bli införstådda för att bilden över hur Use Cases fungerar ska bli tydlig. En av de primära tankarna med UML är att använda sig

av den grafiska bilden, så är det även när Use Cases tillverkas. De grafiska bilderna ska visa hur Use Cases är uppbyggt (se figur 3-2).



Figur 3-2

Här visas ett typiskt Use Case och hur allt kopplas ihop vilket gör att problem som kan missas undanröjs och minimering av framtida problem kan bearbetas innan de dyker upp.

Sample Use Case Diagram (Pacestar Software, 2006)

Fördelen med att använda bilder är att det kan vara lättare att förstå vad någon menar om det går att visa med en bild istället för med en svårtolkad text. T.ex. där det kan finnas språksvårigheter eller kunskapsbrist inom området. Vi ser dem i olika diagram, tabeller

och även när olika organisationer förklaras. Inom reklamen är just bilder ett av de mest använda sätten att påverka människor (Bergström et al., 1999). Bilder är bara en av de delar som bildar ett Use Case och för att skapa en helhet kommer vi här att beskriva mer om andra delar i ett normalt Use Case, samt att gå mer in i detalj på dem.

Intressenter: En intressent är en ägare som har intresse över hur systemet uppför sig. Alla primära aktörer är intressenter. Vissa intressenter har ingen interaktion direkt med systemet. Ägare är vanligtvis någon som tjänar något på systemet som utvecklas eller den som äger det system som utvecklas. Exempel på ägare kan vara direktionen för företaget. Detta lägger kravet på Use Cases att visa hur ägarnas intresse blir säkrat (Cockburn, 2001).

Aktör:

An actor is someone or something outside the system that interacts with the system. (Kruchten, 2003, p.98).

En risktagare kan vara en aktör i ett system. Om det är fallet blir Use Cases mer komplicerade. Detta är dock ett måste om de ska knyta samman Use Cases med användarscenarion (user scenarios). Hur en användare interagerar med systemet är beskrivet i Use Cases.

Tabell 3-1

Ett exempel på en aktörs profil kan se ut så här (Cockburn, 2001)

Namn	Profil: bakgrund och kunskap
Kund	En person på gatan, som kan använda en touchscreen men inte använda Graphical user interface med någon lätthet. Kan ha problem med att läsa eller är färgblind
Godsmottagare	Person som jobbar med mjukvaran ofta och är kanske en sofistikerad användare av systemet. Vill kanske vara med att ändra i användargränssnitt
Chef	Använder systemet då och då, men är inte insatt i några mjukvarusystems funktioner. Otålig.

En aktör specificerar en roll som kan antas av en person, en hårdvara, ett visst datum/tidpunkt, eller en mjukvarukomponent. Varje aktör har ett visst operationellt ansvar som läggs på dem beroende på vilken affärsprocess och vilka affärsregler som de har inom sitt affärsområde. För att uppfylla sitt ansvar kan varje aktör vara tvungen att utföra en rad operationer. En aktör vill att en delmängd av dess operationer utförs av antingen en mjukvaruapplikation eller någon form av hårdvara. Därför är det viktigt att det sätts som mål att systemet ska leverera sådana resultat. Att sätta sådana mål kommer att leda till att den önskade funktionaliteten fås i form av Use Cases. Det är viktigt att varje mål har ett eget Use Case (Guiney et al., 2003).

Användarhandling:

En aktör utför en handling varje gång som han väljer att göra något med systemet. Detta kan vara att få en beräkning utförd eller att få en algoritmisk procedur gjord (Kruchten, 2003). En handling utförs i sin helhet av systemet eller inte alls. Varje handling som en aktör gör med systemet bör ge aktören något av värde tillbaka (Kruchten, 2003). De handlingar som en aktör vill utföra i systemet återges i Use Cases. Normalt är namnet på ett Use Case synonymt med det mål som aktören vill uppnå genom att utföra det som finns i ett Use Case (Kruchten, 2003). De olika Use Cases som finns i ett system representerar de olika sätt som en aktör kan interagera med systemet. Framtagandet av Use Cases fortsätter tills alla handlingar som en användare vill kunna utföra i systemet är täckta. Varje handling är normalt ett separat mål för aktören (Preece et al., 2002).

Handlingsflöde: Vår studie av handlingsflöden är inriktad på de specifika händelseflödena. Dessa beskriver de handlingar som sker mellan aktören och användaren i ett Use Case. Modelleraren försöker att gruppera liknande flöden i ett och samma Use Case (Kruchten, 2003). Det är viktigt att här påpeka att det är två flöden det handlar om, bashandlingsflödet och det alternativa handlingsflödet. Båda skrivs med normal text med de normala termer som finns inom problemområdet som det arbetas i. Hur detaljerat ett Use Case ska vara är ofta något som är en intern företagsangelägenhet. Det samma gäller hur ett företag använder Use Case-modellen och anledningen till att de använder den. Generellt kan det dock sägas att ett Use Case-handlingsflöde tydligt ska beskriva interaktionen mellan aktören och systemet. Jacobson uttrycker det så här

Since the Use Case often focus on a particular functionality of the system, it is possible to analyse the total functionality of the system in an incremental way. (Jacobson et al., 1993, s. 163)

Vilken detaljnivå som ett Use Case ska ha är kraftigt beroende av vad det ska användas till och vilka mål som ett företag vill uppnå med det. Målet med ett Use Case leder till interaktioner med ett system för att det ska gå att leverera detta mål. Beskrivningen för att nå fram till målet i ett Use Case består av två delar, basflöde och alternativa flöden. Med basflöde menas det flöde där allt går väl i interaktionen mellan aktörer, systemet och användaren. I alternativa flöden hanteras alla andra fall, så som ovanliga kombinationer som uppstår, flervalsmöjligheter, olika sätt att interagera med systemet, systemåterställning, felhantering och liknande. Alternativa flöden kan ofta uppstå för att hantera ett specifikt affärsförhållande. Det är när det upptäcks att basflödet inte täcker in något, som det skapas ett alternativt flöde från den punkten i basflödet. En sådan förgrening från basflödets interaktions flöde kallas anknytningspunkt (extension point)(Kruchten, 2003).

Handlingsbasflöde: Basflödet beskriver huvudflödet av handlingar som finns i ett Use Case. Basflödet ska innehålla det scenario där användaren interagerar på ett perfekt och korrekt sätt med systemet. När en modellerare har skrivit ett basflöde så inser han snabbt att det finns många alternativa sätt att interagera med systemet. Dessa olika vägar som inte är en del av basflödet hanteras i alternativa flöden (Kruchten, 2003).

Alternativa flöden: För att beskriva det alternativa flödet så är det i den här delen av ett Use Case som modelleraren hantera exceptionella eller udda händelseflöden. De olika händelseflödena ska så småningom beskriva de flesta av de möjliga sätt som Use Cases behandlar. Från början bör en modellerare dock inte lägga ner för mycket arbete på de alternativa flödena, då det är väldigt vanligt att det initialt ofta ändras i basflödena i olika Use Cases. Att då ha lagt ner mycket arbete på alternativa flöden som ändå kommer att kastas bort är slöseri med resurser (Jacobson et al., 1993). Det gäller också att inse att det normalt inte går att täcka in alla alternativa flöden då det i ett större system helt enkelt finns allt för många av dem. Ett alternativt flöde kan ses som en alternativ historia. Det är som att ha en tidslinje som helt eller delvis ersätter basflödet eller ett annat alternativt flöde. När ett alternativt flöde en gång förgrenats från basflödet, så kommer det aldrig att flyta samman med det eller något annat flöde igen. Detta är uppenbart då det i ett Use Case inte ska finnas mer än en tråd som styr kontrollen i det.

Anknytningspunkt: Som vi beskrivit tidigare så är en anknytning bara en del av ett Use Case. Det går att se på varje anknytning som om den vore ett litet Use Case som hänger samman med basflödet i ett huvud- Use Case. Varje anknytning är dock abstrakt eftersom de inte kan instansieras separat. Det är alltid så att det är något annat Use Case som behöver dem, normalt ett konkret eller genomgående Use Case (Jacobson, 2006). Den specifika punkt som kallas anknytningspunkt kan ses som ett sätt att tillföra extra beskrivande text till ett utökat Use Case (Kruchten, 2003). Anknytningspunkter används för att skilja ut beteenden som är vanliga i många Use Cases. Det bör göra att den färdiga Use Case-modellen blir lättare att överblicka och arbeta med.

3.4 Vad menas med konceptet spårbarhet?

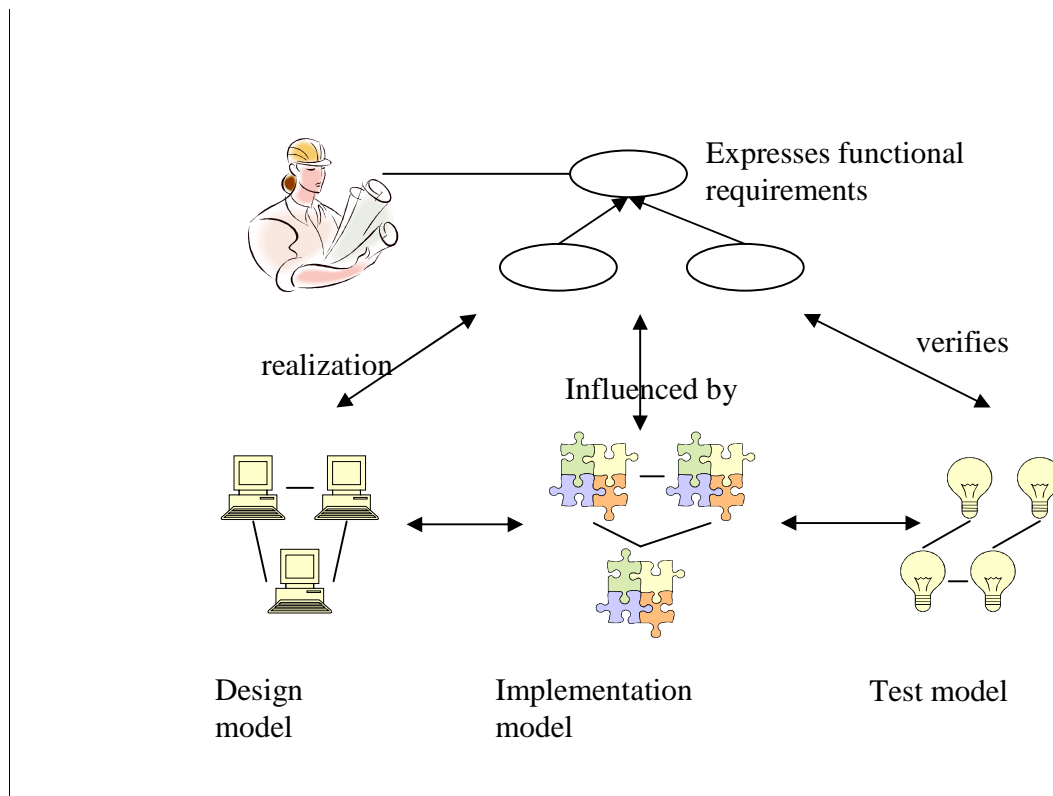
En av de viktigaste delarna, som behövs för att ha en möjlighet att kunna återanvända Use Cases, är spårbarhet vilken är viktig även utan återanvändning. Spårbarhet har definierats som:

Any textual, or model item, which needs to be explicitly traced from another textual, or model item, in order to keep track of the dependencies between them. (Spence, 2000, s. 1)

Med detta menas att det ska vara möjligt att spåra en viss funktionalitet eller ett objekt genom ett helt projekt. Funktionaliteten eller objektet ska refereras till på samma sätt genom hela projektet. Det finns två sorters spårbarhet, underförstådd och uttalad (Spence, 2000). Underförstådd spårbarhet kan uppnås genom att ha namnkonventioner så att ett namn inom ett projekt alltid refererar till samma objekt. Ett annat sätt är att markera objekt i olika modeller på samma sätt så att det går att lätt känna igen ett visst objekt i de olika modeller som används (Jacobson 1997).

Till exempel så går det i RUP:s ramverk se att de spårar implementeringen i designmodellen tillbaka till det Use Case som implementerats. Här finns då möjligheten

att kunna skapa namnkonventioner för återanvändning. Ett annat sätt som går att använda är att skapa olika vyer för att illustrera hur krav i en modell möts i en annan modell så att de olika delarna bildar en helhet. Detta görs för att det enkelt ska kunna gå att se att implementeringen av ett Use Case i designmodellen och de andra modellernas delar samarbetar för att ett helt Use Case ska kunna implementeras. Detta ger då en hjälp att beskriva hur Use Cases ser ut vilket kan vara till stor hjälp vid en förklaring av hur delarna kan återanvändas vid senare tillfällen. Att arbeta på detta arbetssätt kallas att använda sig av ett Use Case-drivet arbetssätt.



Figur 3-3

Use Case-modell: Här visas de olika modellerna och de underförstådda förhållanden som finns mellan dem. Detta för att förtydliga hur den underförstådda spårbarheten påverkar utvecklingsprocessen när ett Use Case-drivet arbetssätt används. (Spence, 2000)

Den underförstådda spårbarheten är begränsad till de relationer som är tillgängliga i den modellnotation som används. På grund av detta så behövs det en del komplementär delar, uttalad spårbarhet, när ett Use Case-drivet arbetssätt används. Dessa behövs för att få en fullständig kravspecifikation för den mjukvaran som ska utvecklas. I RUP kallas dessa för Tilläggspecifikationer (Supplementary Specifications), Ordlista (Glossary), Arbetsfallet (Business Case) och Visions (Vision) dokumenten, vilka innehåller de behov, mål och de kännetecknande delarna som behövs i projekt. Dessa dokument visas inte i figuren ovan (se bilagan Business Process Model).

Den uttalade spårbarheten används för att täcka in de områden som inte täcks in med den underförstådda modellnotationen.

I RUP så är Tilläggspecifikationerna exempel på uttalad spårbarhet, i dessa går det att ta med sådana krav som inte täcks in i modellnotationen. T.ex. vilken känsla som det är tänkt att en hemsida ska förmedla till sina kunder.

Ordlistan, där förkortningar, begrepp och akronymer som används i projektet förklaras så att alla användare ska ha samma uppfattning om vad de betyder. Den förstärker och klargör dessutom vilka namnkonventioner som ska användas i projektet.

Arbetsfallet, där det förklaras hur det som produceras ska passa ihop, som ett huvudscenario, med det övriga arbetet som bedrivs på t.ex. den firma som producerar Use Cases. Det är även här som utvärderingen av huruvida ett projektet är lönsamt att satsa på finns. Huvudmålet med arbetsfallsdokumentet är att det ska gå att skapa en ekonomisk plan för hur visionen i visionsdokumentet ska kunna realiseras (Probasco, 2000). I Visionsdokumentet beskrivs hur visionen ser ut. Det beskrivs också hur det är tänkt att den ska uppfyllas, vilket sker genom att implementera de Use Cases som skapats.

Visionsdokumentet kan även vara ett sätt att knyta ihop flera Use Cases för att klargöra meningen med dem och ge en klarare helhetsbild av vad det är som Use Cases ska bidra med i det system som håller på att skapas. I visionsdokumentet ska det finnas en analys av det problem som ska lösas. Det ska dessutom kunna klargöra risktagarens behov, och det är också i visionen som systemet definieras. Många gånger kan visionsdokumentet vara skrivet nästan i kontraktsform. Normalt är det detta dokument som styr stora delar av hur ett projekt utformas (Probasco, 2000).

Alla dessa dokument är exempel på uttalad spårbarhet som inte finns med i modellnotationen (Spence, 2000). Det vi vill visa är vikten av spårbarhet och vid möjlig återanvändning behövs båda dessa typer av spårbarheter.

4 Telecas form av Use Cases

För att kontra vår litteraturgenomgång behöver vi presentera Teleca, där vi gjorde vår undersökning, och hur deras bild är av Use Cases. Vi kommer att beskriva vår situation på Teleca under denna tid. Dessutom måste det förstås, att vi var inbjudna till det här projektet. Vi tvingades att inse, att angående personal, tid och verktyg så fanns det en gräns, för vad Teleca kunde erbjuda. Vi kunde inte begära att deras projekt skulle stanna upp för vår undersökning. De gav oss trots allt möjlighet att studera användandet av återanvändbara Use Cases på en praktisk nivå, samt att genomföra vissa intervjuer med kontaktpersonen och vissa andra projektmedlemmar inom organisationen. Vi kommer att presentera hur Teleca skapar, hanterar och anser att deras Use Cases ska vara, samt vad det står för inom deras organisation.

4.1 Teleca som företag

Teleca är ett företag som i grunden är en konsultgrupp specialiserad på mobiltelefonsteknologi. De bygger moduler till mobiltelefonin under hård konkurrens i en marknad, som är under ständig förändring, och där tiden är en av de viktigaste faktorerna. Nya implementeringar sker hela tiden inom denna marknad. För att kunna hänga med i dessa snabba vändningar, skaffade Teleca in ramverket RUP, från Rational vars ägare är IBM. Därigenom skapades snabbare genomloppstider för projekten samtidigt som systemet väl passade in i organisationen. Resultatet var väldigt gott, men de ville förbättra det ytterligare, och det var här återanvändningsbara Use Cases kommer in. Det var i detta läge, vi fick en chans att göra en undersökning på Teleca angående Use Cases. De fördelar, som Teleca ser, med möjligheten att kunna återanvända Use Cases, är en av de huvudsakliga anledningarna till våra studier i det här ämnet och de möjligheter, som det bjuder på. Enligt vår kontaktman inom företaget förväntar de sig bland annat tidsvinster genom kortare utvecklingstid vid skapandet av mjukvara till mobiltelefoner, eftersom vissa funktioner, som till exempel själva telefonfunktionen, återskapas hela tiden. Att ta fram nya Use Cases för detta varje gång är ett tidsödande arbete, som kostar pengar. För att minska tid och kostnader i deras projekt provar de nu återanvändningsbara Use Cases. Den beräknade minskningen av tiden ger dem även möjligheten att lättare uppnå ett av sina huvudmål, som är att få ut ny mjukvara fortare på marknaden, helst fortare än deras konkurrenter. De ser fördelen med snabbare Use Cases, som ett medel att distribuera dem fortare till sina olika avdelningar för en djupare analys. Dessa kan då börja bygga sina moduler i ett tidigare skede, än vad de skulle ha kunnat annars, och de hoppas att det ska kunna leda till att den totala projekttiden skall kortas

ner, och att personalen därmed ska kunna bli tillgängliga för andra projekt tidigare. De anser också att återanvändning av Use Cases har den fördelen, att de är lättare att anpassa till en specifik kunds behov.

4.2 Modulverksamhet

Mobiltelefonin bygger mycket på moduler, där vissa moduler i huvudsak fungerar på samma sätt hela tiden. Det är bara nya delar som infogas men inför kunden måste hela telefonen visas. Eftersom det är så, räknar Teleca med att tjäna tid på att det finns färdiga Use Cases på vissa moduler.

För att se lite mer objektivt på Use Cases så sökte vi upp personal på ett företag inom en annan bransch, som inte arbetar med moduler utan skapar applikationer. I vårt fall så hamnade vi på ett företag inom industriområdet. Anledningen till att vi valde dem var för att se hur de hanterar Use Cases och hur de hanterade återanvändbarhet. Vi valde en Application Manager på Alfa laval AB eftersom han är den personen som ansvarar för Use Cases på ett visst system. Han tog sig tid med en kortare intervju om hur de handskas med Use Cases. Det industriella företaget Alfa Laval AB bygger flertalet av sina egna system. Det visar sig att det är bara fåtalet av de egenskapade systemen som har en stomme med Use Cases och att de använder ett egentillverkat ramverk. Där framkom att i de systemen sker det ommodellering istället för att återanvända. Use Cases skulle inte ge den effekten eftersom det inte sker i modulplan.

Det intressanta här var deras tillämpning av Use Cases inom beräkningssystemet Computer Aided Sales (CAS). Där de inte återanvänder Use Cases utan ommodellerar dem istället. Själva Use Case-modellen liknar RUP:s därför var det intressant att se hur det går till att ommodellera dem. Detta hjälpte oss att se skillnaderna mellan ommodellering och återanvändning.

Det var något som vi kunde dra nytta av när vi skulle jämföra det som kallas ett standard Use Case enligt RUP med ett Use Case som Teleca anser vara ett återanvändningsbart. Med all den informationen om Telecas arbetssätt och dokument angående deras produktutveckling fick vi en inblick i både vanliga Use Cases och de som är möjliga för återanvändning. Telecas mål har varit att skapa återanvändning för att telekomindustrin bygger på principen att nya moduler blir nya projekt. Dessa moduler blir inbyggda i nya modeller av mobiltelefoner. Med återanvändningsbara Use Cases kan undersökningstiden minimeras i jämförelse med att göra nya Use Cases. Återanvända Use Cases kan ändå fylla den fulla funktion som de är ute efter och detta ger då Teleca ett försprång gentemot deras konkurrenter. Men det är viktigt att notera skillnad mellan de Use Cases som återanvänds och de som ommodelleras. För idag redigeras Use Cases ofta inom projekt, särskilt projekt med system som vidareutvecklas. Detta bekräftas av application managern på Alfa Laval, som berättar om ett projekt kallat CAS, vilket innefattar ett beräkningsprogram som skapats med hjälp av Use Cases. När utveckling sker redigeras Use Cases om för att kunna innefatta nya funktioner eller rent grafisk utveckling. Det är vad vi kallar för modellerade Use Cases. För Telecas del är återanvändningsbara Use Cases till stor nytta, men för att det ska vara meningsfullt, så måste tiden det tar att skapa

dem vara mindre, än den tid det tar att anpassa dem till det, som är ett nytt huvud-Use Case. Om det inte är på det viset, så skulle det innebära, att tidsvinsten blir marginell, och då tappar konceptet återanvändning sin mening. Teleca anser, att med ommodellering finns risk för, att det tar längre tid, än att skapa ett nytt Use Case.

4.3 Telecas syn på möjligheten att göra ändringar på Use Cases

Vi förstod genom våra intervjuer att företaget är i en fas där de provar ut effektiviteten med att ha återanvändningsbara Use Cases. I bakgrunden har de RUP, som de vill vidareutveckla till ett effektivare verktyg än vad det är idag. Detta har krävt att de har ändrat vissa bitar i sina metoder och strategier, men det kan diskuteras, om de uppnår målet och ändå kan säga sig följa RUP.

En del var att de ansåg att för återanvändande av Use Cases, var de tvungna att äga förundersökningarna, vilket de ofta gör. Ägandet innebär, att när de har skapat ett Use Case, så kan det arkiveras för återanvändning, vilket annars inte vore möjligt. Äganderätten ger en viss aspekt på vilka krav, som kan behövas, för att möjligheten att återanvända Use Cases ska kunna bli reella.

Med Telecas syn på fungerande Use Cases kommer det in delar, som inte tas upp av dagens Use Cases. Bland annat finns en annan syn på spårbarhet och arkivering. En förutsättning är, att spårbarheten även inkluderas på ett sätt, som gör det lätthanterligt att finna rätt Use Case till rätt projekt, så att det bara kan stoppas in som ett klipp och klistra objekt. Detta ska sedan kunna levereras vidare direkt till andra berörda delar av projektgrupperna, för att de ska kunna arbeta vidare med projektet och bearbeta det, precis som om det skrevs unika Use Cases. Detta medför att det måste finnas ett arkiv, som kan användas på samma sätt som att låna en bok på ett bibliotek. Idag finns stommen där, men den behöver utvecklas mer.

När det gäller begrepp och delar av Use Cases så är Telecas synsätt att de inte kan koppla dem mot några fysiska objekt eller mot andra Use Cases. Om det inte finns några bindningar till gränssnittet kan de med User scenarior modellera för tekniker som kan uppstå i t.ex. röststyrning eller någon annan teknik som kan komma i framtiden. Detta har de uppnått med sitt nya tankesätt, men den här typen av Use Case-tänkande behöver utvecklas hela tiden.

4.4 Telecas Use Cases

Teleca håller med om att Use Cases är väldigt användbara men de vill ta dem ett steg längre genom att återanvända dem vilket de inte gjort tidigare.

I en intervju med vår kontaktperson berättar han att idag har Teleca två typer av Use Cases. Ett som är på utvecklingsstadium och ett som anses som ett standard-Use Case. Det som är en standardmodell är taget direkt från Rationals manualer, vilka inte bygger på återanvändbarhet utan lägger tyngden på den konventionella typen av Use Cases, som Ivar Jacobson beskrivit i sin modell inom RUP. Detta är samma princip som vi beskrivit i litteraturgenomgången. Sedan har de en modell för möjligheten att kunna återanvända och den har ändrats så att Use Case-modellen ska passa till deras syfte, men ändå anpassat sig efter RUP-modellen.

När de beskriver återanvändbara Use Cases finns det bara marginellt beskrivet om hur de fungerar. Det beror till viss del på att de är under utveckling, men vi har fått det förklarat för oss som att RUP:s manual är grunden även för de Use Cases de vill återanvända.

Genom våra observationer har vi kunnat se att dokumentationen från Teleca visar deras sätt att använda Use Cases inom standardiserade Use Cases. Bland dokumentationen fanns bland annat Rationals dokumentation men även "Use Case Report" som är Telecas sätt att standardisera personalens dokumentering av varje del i ett Use Case.

Dokumentationen för återanvändbarhet är, som tidigare påpekats, fortfarande på ett utvecklingsstadium och därför begränsad. Till stor del ligger den dokumenteringen i en fas för sammanställning tillsammans med vissa delar av vårt material.

Genom andra intervjuer fick vi fram vissa viktiga delar för Use Cases som ska återanvändas. Ett exempel på det är spårbarhet som i första hand är till för att kunna hålla dem inom en ram där de lätt ska kunna hitta dem igen. I många delar av den spårbarhet som Teleca försöker använda inom återanvändande, är det så att de försöker efterlikna vanliga Use Cases eftersom de fortfarande vill hålla sig inom RUP standarden.

Observationerna gav oss även möjlighet att titta på ett äldre projekt där de använder sig av vanliga Use Cases. Där fick vi en djupare förståelse för hur vanliga Use Cases fungerar i ett operativt projekt. I det projektet undersökte vi hur Use Cases hade namngivits, hur texten i Use Cases var skriven och hur de fanns arkiverade.

Rubriceringen av Use Cases från arkivet i det äldre projektet ser annorlunda ut jämfört med återanvändbara Use Cases (se bilaga *Reusable Use Case image Teleca* och bilaga *Use Case image Teleca*).

Enligt Telecas representant ger de olika Use Cases en bild över deras två olika Use Case-modeller.

Vid återanvändning krävs ett nytt tänkande, enligt deras representant, angående hur arkivering av dem ska ske, så att de är lättillgängliga till andra projekt. I dagsläget arkiverar varje projekt sina Use Cases separat inom projektet. De anser att för själva arkiveringen måste företaget tänka på att det kan vara t.ex. en röststyrd enhet och då finns inte något behov av knappar att trycka på. En stor del av deras lösning är själva arkiveringen där tar man upp vilken styrning telefonen har genom sin namnkonvention. Namnkonventionen beskriver att det är ett Use Case och vilken del av projektet som det tillhör och vilken typ av modul det är. Därefter kommer vilken revidering av vilket Use

Frost & Hellström- Återanvändbara Use Cases inom telekomindustrin

Case det är och sist vad det berör. Teleca menar att genom att ha rätt rubricering går det att skapa ett fungerande arkiv för återanvändbara Use Cases

5 Diskussioner och resultat

I detta kapitel tar vi upp våra resultat. Dessa kommer vi att diskutera och knyta samman med litteraturgenomgången och vår undersökning. Vi kommer också att belysa vilka användningsområden som den här typen av Use Cases kan vara intressanta för samt vilka viktiga skillnader som vi hittat mellan ett återanvändbart och ett icke återanvändbart Use Case. Vi behandlar också varför det är så viktigt med spårbarheten för återanvändbara Use Cases och vikten av en sund form av arkivering. Slutligen kommer vi att ta upp vilka vinster som kan förväntas att uppnås med återanvändbara Use Cases. Vi avslutar kapitlet med en enkel resumé över det som vi ser som de största skillnaderna mellan de olika Use Cases.

5.1 Vår diskussionsgrund

Vi har uppfattat att återanvändbara Use Cases kan vara ett användbart verktyg om de har rätt uppbyggnad, vilket kan vara svårt att uppnå. Vi anser att rätt uppbyggnad är när man kan återanvända ett Use Case utan att behöva ändra i det. Det borde då gå att tjäna tid, personal och pengar på att använda dem. Anledningen till att det kan vara svårt att få rätt uppbyggnad på återanvändbara Use Cases är att det finns nackdelar som t.ex. att det initialt kan vara svårt att administrera. I början krävs utbildning på hur Use Cases ska skrivas, så att de blir möjliga att återanvända. Detta blir särskilt tydligt, om det inte finns någon klart angiven standard för hur Use Cases och dess delar ska dokumenteras. Från början trodde vi att det fanns möjligheter inom telekomindustrins olika områden att använda återanvändningsbara Use Cases. Vi har senare i vår undersökning genom tolkning av resultaten kunnat konstatera att detta är möjligt under rätt förutsättningar som exempelvis när det skapas systemmoduler och med viss modifiering vid tillverkandet av Use Cases.

Vi trodde också från början att sättet att dokumentera Use Cases på skulle skilja sig från sättet som finns beskrivet i litteraturen. Under våra observationer fann vi snart att det i Telecas fall är så. Det sätt som beskrivs i "Use Case Report", som var en del av Telecas dokumentation, var inte det som Teleca var ute efter när det gäller återanvändning. De kommer att behöva skriva om delar av dokumentationen för att de ska uppnå sitt mål med återanvändning. Detta eftersom "Use Case Report" beskriver normala Use Cases utan att ta hänsyn till återanvändning. Tonvikten och skillnader med dokumentationen kan bli både namnkonventioner och hur de rent faktiskt ska skrivas. Vad vi även kunnat konstatera när vi studerat Telecas återanvändningsbara Use Cases är att strukturen på Use

Cases är den samma som i normala Use Cases men innehållet i de olika delarna och framför allt formuleringen av texten i dem är annorlunda. Annorlunda på det viset att hänvisningar till gränssnittet är helt uteslutna i flödena i Telecas återanvändningsbara Use Cases. De använder sig också av ett mer generiskt skriftspråk, när de redogör för de händelser som Use Cases beskriver.

Under våra observationer kunde vi också konstatera att de Use Cases, som var enklast att återanvända var de, som inte hänvisade till gränssnittet i den applikation som det Use Case ingick i. Anledningen till att det var enklare att återanvända dessa var att det vid återanvändandet då inte behövdes tas hänsyn till eventuella begränsningar som det tidigare gränssnittet kunde innehålla. Detta oavsett om det är ett grafiskt gränssnitt på en datorskärm eller ett fysiskt gränssnitt på till exempel en maskin. Om ett Use Case innehöll många hänvisningar till gränssnittet så blev det betydligt svårare eller rent av omöjligt att använda det igen, om inte gränssnittsmiljön som det skulle användas i var identisk med fallet som Use Case använts i tidigare. Ofta är så inte fallet.

Ytterligare en skillnad som vi kunnat visuellt observera är att i och med att det inte hänvisas till gränssnittet så blir namngivningen av Use Cases i Telecas fall annorlunda. I ett normalt Use Case så innehåller namnet ofta en referens till gränssnittet (se Use Case image Teleca), men i Telecas återanvändningsbara Use Cases så finns inte detta (se Reusable Use Case image Teleca), istället så baseras namnet enbart på den funktionalitet som Use Cases löser. En stor fördel som de såg med detta på Teleca är att de kan arkivera sina Use Cases enbart med tanke på vilken funktionalitet de löser och inte med tanke på funktionalitet och gränssnittet. Att inte behöva ta hänsyn till gränssnittet gör då att Use Cases inte behöver lagras om och om igen. Det räcker att lagra Use Cases en gång även om funktionaliteten som det löser används många gånger. Varje gång som ett Use Case behövs med sin funktionalitet så är det bara att återanvända det Use Case som lagrades istället för att skapa ett nytt. Detta sparar då utrymme, i och med att det inte behöver arkiveras lika många Use Cases.

En annan och nog så viktig fördel är att det blir betydligt färre Use Cases att söka igenom när det är dags att hitta det Use Case som det är dags att återanvända.

Det samma gäller när det är dags att kontrollera om det finns ett Use Case som det går att återanvända eller om det måste göras ett nytt.

Vi kunde också observera, att sökandet efter ett Use Case underlättades, när det fanns en standardiserad namnkonvention, så som de har på Teleca eftersom alla på företaget då använder samma sätt att namnge sina Use Cases, och därför väl känner till den lagringsstruktur som används. Genom namnkonventionen får de också en bibehållen spårbarhet i de projekt, som de återanvändningsbara Use Cases ingår i. En förutsättning är förstås, att det används samma namnkonvention i de projekt, som Use Cases ska ingå i framöver.

5.2 Användningsområden för återanvändbara Use Cases

Vid återanvändning av Use Cases krävs det en speciell marknad om det ska löna sig. Det måste finnas ett behov av att använda sig av samma funktionaliteter vid upprepade tillfällen. Ett av dessa områden är utan tvekan inom ett företag, som säljer produkter i modulform som exempelvis Teleca. Det är en fördel om företagets produkter har samma bas, innehåller samma moduler, eftersom det då är lättare att återanvända Use Cases från tidigare projekt, som i detta fall där det arbetas med samma produkt men vidareutvecklingen av den. I andra fall finns möjligheten att ommodellera Use Cases, en möjlighet som kan ge nästan samma resultat. En ide som Teleca hade när det gäller återanvändning av Use Cases är att komplettera dem med user scenarios, typ som i Extreme Programming där man skapar enklare historier som bas för produkten, för att på så sätt kunna göra Use Cases snabbare. Idén är att försöka skapa Use Cases som direkt kan läggas till som funktionalitet utan att skriva om de gamla Use Cases. På detta sätt kan de med den tidsvinst som det innebär att inte behöva skriva nya Use Cases skapa program snabbare och effektivare än deras konkurrenter.

5.3 Skillnader mellan viktiga begrepp i ett normalt och ett återanvändningsbart Use Case

Ett av de viktigaste begreppen inom denna verksamhet är krav och vad de innebär både för kund och för tillverkare. Styrkan i Use Cases är att hitta och täcka in så många av kundens krav som möjligt för att säkerställa att kunden får den slutprodukt som kunden vill ha.

I det fall som vi jämfört med, Teleca, så har kunden normalt redan gjort en kravanalys på vad som behövs. Sen är det upp till tillverkaren att förverkliga dem och visa upp möjligheter på lösningar för kunden. Det är här som Use Cases kommer in i bilden. När en lösning har godkänts skickas den ut till de olika avdelningarna i form av Use Cases, som de sen förverkligar.

När vi ser på återanvändningsbara Use Cases så kan nya utvecklare som använder sig av Telecas mall känna sig lite förvirrade i fråga om hur kraven skall användas, detta på grund av att de inte ska koppla dem till andra Use Cases så som det görs i vanliga Use Cases.

Telecas sätt att se på krav är att de ser dem som en begränsning eller så lägger de in dem som en beskrivning i sina Use Cases. I deras Use Case-dokumentation så finns det ingen nedskrivna del där de samlar sina krav. Men för att kunna göra en jämförelse i den här uppsatsen så tittar vi på hur de agerar mot **Funktionalitet, Användbarhet, Pålitlighet, Prestanda och Support**.

Ett krav kan visas i mer än en dimension tack vare att de kan beskrivas grafiskt och i text. Detta gör att styrkan i Use Cases ökar. Kravspecifikationen med de funktionella kraven kan förstås av fler personer som kommer att använda den för att skapa den första förändringen som systemet eller modulen kommer att genomgå under modelleringen.

Vi observerade att på Teleca har de en annan terminologi för icke funktionella krav. De kallar dem för speciella krav. De ser på sina speciella krav som något, som inte täcks in i deras handlingsflöden, utan snarare något som kommer att påverka deras designmodeller utan att vara en del av dem.

I en av våra intervjuer så fick vi reda på att de på Teleca anser, att i mobiltelefonbranschen så är de icke funktionella kraven en av de viktigaste aspekterna, som de måste ta hänsyn till, när de ska utveckla nya mobiltelefoner, eftersom det ofta är genom de icke funktionella kraven, som de kan få nya idéer om vad som ska skapas härnäst. De begränsar också vilka funktionaliteter som är möjliga att skapa och vad som inte går att skapa. Med **funktionalitet** så menar vi de operationer som systemet tillhandahåller vilka är det minimum som modulen/systemet måste tillgodose.

När det gäller Teleca så talade personalen, under våra intervjuer, om för oss att om det ska gå att göra ett Use Case återanvändbart så är det viktigt att vara noga med hur den funktionella delen av dem skrivs. Detta då det är en av de delar som avgör om det är möjligt att göra ett Use Case återanvändbart. De förklarade att om det skrivs på fel sätt i den här delen så kommer det inte att gå att återanvända. Den funktionella delen måste skrivas med återanvändning i åtanke. Detta är särskilt viktigt i Telecas fall eftersom många av deras applikationer(moduler) startar på samma sätt och därmed är möjliga att återanvända om de skrivs på rätt sätt, vilket i Telecas fall innebär att inte ha hänvisningar till gränssnittet. De måste också tänka på vilka funktionalitetskrav, som ett Use Case uppfyller, när det ska arkiveras, så att deras namnkonvention kan upprätthållas. Detta måste särskilt uppmärksammas av personer, som är vana att skriva vanliga Use Cases.

På Teleca så anser de att **användbarheten** av ett korrekt skrivet Use Case blir mångdubbelt större om de kan återanvända det. Om det är samma kund som skall använda samma Use Case så behöver de inte förklara det lika ingående. Detta är något som de på Teleca anser att de borde kunna spara både tid och pengar på. Både åt sina kunder och det egna företaget.

Vi hade en idé om att det bör skapas någon form av huvud-Use Case som håller samman de olika Use Cases som finns. Ett huvud-Use Case kommer inte att gå att återanvända. För att förankra våra idéer så tog vi några intervjuer med vår kontaktperson på Teleca om vilka perspektiv de hade om återanvändbarhet och ommodellering. Under de intervjuerna klargjorde han för oss vad Teleca anser om återanvändningsbara Use Cases. Idag skrivs redan en form av huvud-Use Case, men det är mer som scenarion, men samtidigt poängterade han då vikten av att det gjordes en tidsvinst vid återanvändandet, detta för att funktionaliteten skulle vara lönsam och berättiga existensen av återanvändningsbara Use Cases.

En annan viktig aspekt som de påpekade för oss på Teleca är att för att det ska löna sig att återanvända ett Use Case så måste **pålitligheten** i ett Use Case vara hög. För den kritiska delen med återanvändning är att när någon ändrar Use Cases, som de gör på Teleca, finns risken att de tappar bort bitar som klart och tydligt skulle visas i Use Cases som skrivs på det vanliga sättet. Unikum för varje Use Case finns inte när det är återanvändbara Use Case som används, de kommer att skrivas för flera projekt. Med det menas att de inte skrivs för en enskild kund eller produkt utan för multipla projekt. Multipelt tänkande kan

ge fördelar i början men tappar kraft i ett långt perspektiv. Ett Use Case kan bli utnött och det bör då arkiveras och ersättas med ett nytt som passar in i organisationen. Då är det viktigt att det finnas någon som kan ta det ansvaret och se till att det blir gjort. Detta så att återanvändningen inte för in kända problem i ett nytt projekt. Ett väl beprövat och fungerande Use Case behöver betydligt mindre **support** och risken att fel uppstår i systemet minskar. För att supporten ska bli så bra som möjligt så bör det läggas ännu mer vikt på att Use Cases är rätt skrivna och väldefinierade från början.

Att återanvända ett Use Case med samma funktionaliteter kommer att bidra till att förutsägbarheten kommer att öka och den **prestanda** som uppnås kommer att bli mer förutsägbar, vilket är en fördel för både modellerare, programmerare samt den tilltänkta användaren om de har använt ett system baserat på samma Use Case tidigare. Det blir då lättare för kunderna att ta till sig funktionaliteten som de känner igen sen tidigare och detta gör att **användbarheten** av ett Use Case ökar för både kunder och Teleca. Detta är av stor vikt inom Telecas bransch där kunderna ofta byter en mobiltelefon mot en annan. På Teleca kommer de att behöva införa nya regler för hur Use Cases ska skrivas om det ska gå att återanvända deras Use Cases. Det i sin tur kräver utbildning av personalen till den standard som krävs av dem som ska använda dem, som exempelvis namnkonventioner och kopplingar till andra Use Cases. Vi vill poängtera att en av de viktigaste reglerna är att de återanvändbara Use Cases byggs så att det inte finns något samröre med andra Use Cases eller den teknologi som används. Exempelvis kan de inte referera till någon specifik knapp, för nästa gång ett Use Case används så finns kanske inte knappen som det refereras till. Det som utfördes genom att trycka på knappen kanske ska utföras med ett röstkommando istället.

5.3.1 Use Cases olika delar

När vi studerade de olika delarna i ett Use Case så insåg vi att ett återanvändningsbart Use Case inte skulle vara beskrivet på exakt samma sätt som i ett standard-Use Case. Vi tog fasta på var de synliga skillnaderna fanns och hur de olika Use Cases utmärkte sig. Om vi börjar med den grafiska delen så bygger Use Cases på att det på ett lätt sätt ska kunna visa de kravställningar kunden har satt upp och att de är möjliga att realisera. Många av dem som använder Use Cases idag har svårt att se fördelen med grafiska Use Cases. Det är inte alla som tänker på att en kund som inte är så insatt i de tekniska termerna kan få en ökad förståelse för Use Cases när de visas i bildform. Det kan även finnas vissa språkliga problem som kan bli lösta och förtydligade av att Use Cases visas i grafiska notationer. Fördelen när de återanvänds är att den grafiska bilden inte är nödvändig att rita på nytt när återanvändningsbara Use Cases används.

När vi studerade hur Use Cases skiljer sig vad gäller aktörerna så kunde vi observera att de på Teleca ser en aktör som en vital del i projektet eftersom en aktör kan till exempel representera en person som ska använda en av deras produkter. Utifrån dem så gör de en projektanalys med användarscenarion. De gör detta för att säkerställa att de har uppfyllt alla krav som en kund vill ha. Precis som i RUP:s Use Cases så kan det i Telecas fall vara så att en aktör kan vara en hårdvarukomponent, ett specifikt datum/tidpunkt, eller en

mjukvarukomponent. Use Cases kommer att visa hur funktionaliteten i systemet kommer att fungera mellan användaren och systemet allt eftersom det fortskrider. Som nämnts tidigare så har varje aktör särskilda ansvarsområden beroende på hur affärsprocessen och affärsreglerna ser ut inom affärsområdet. En skillnad som vi uppmärksammade på Teleca var att det ofta är så att deras kunder har valt de aktörer som ska ingå i projektet redan innan de börjar göra Use Cases.

De olika flödena fungerar i stort sett likadant men hur de på Teleca ser på flödenas effekter kan skilja sig en del. I ett huvudscenario bör modellaren förklara var scenariot börjar och var det slutar, om ett Use Case skall återanvändas visas det bara hur/varför det startar eller slutar. I ett återanvändningsbart Use Case får inte finnas några navigeringsbara beskrivningar. Vad vi menar med navigeringsbara beskrivningar är, att ett Use Case inte får härleda till något annat Use Case för att inte förlora både effekten och syftet, om det skall återanvändas. Det får inte finnas anknypningspunkter mellan olika Use Cases, då det vid en sammanbindning försvinner en del av återvinningsmöjligheten med Use Cases. Anledningen till detta är, att de blir beroende av varandra, och därmed måste inte bara ett utan flera Use Cases passa in i ett nytt projekt, för att de ska gå att återanvända. Om det ska gå att återanvända så många Use Cases som möjligt, är det alltså klokt att undvika att ha anknypningspunkter till andra Use Cases.

5.4 Vikten med spårbarhet inom Use Cases

Under våra dokumentationsstudier kunde vi observera, att spårbarheten är av största betydelse när ett Use Case ska återanvändas. Det är inte bara för att det ska gå att spåra funktionalitet eller objekt genom ett projekt, utan även för att det ska gå att spåra dem genom flera projekt, därför ökar vikten på att de refereras på samma sätt hela tiden. Inom projekten har både uttalad och underförstådd spårbarhet sin betydelse, då dessa är de viktigaste länkarna till hur Use Cases kan arkiveras. På detta sätt går det snabbt att finna de Use Cases som kan få ny tillhörighet i andra projekt.

Precis som i vanliga Use Cases så är det viktigt att kunna spåra till andra delar av modeller inom ramverket när återanvändbara Use Cases används (se figur 3.3). Detta innebär att det även här går att använda ett Use Case-drivet arbetssätt men de komplementärare Use Cases ökar i värde när det gäller att återanvända.

Vid återanvändning så är det en fördel, enligt Telecas personal, att först skapa ett så kallat huvud Use Case där det går att lägga in gamla Use Cases direkt, och ur dessa går det sedan att få en fullständig kravspecifikation för det projekt som håller på att utvecklas.

När det gäller tilläggspecifikationer så faller de samman med vanliga Use Cases medan i Arbetsfallet så gäller det att ta hänsyn till lönsamheten och utvärderingen.

Utvärderingen är beroende av hur bra de tidigare Use Cases skrivits. Use Cases kan bli påverkade om de inte har en bra stomme på sina återanvändningsbara Use Cases. Om det visar sig att stommen på Use Cases inte är bra finns risken att hela idén med projektet fallerar. Detta eftersom lönsamheten påverkas om inte utveckling av nya Use Cases kan minskas.

När det gäller Visionsdokument så blir det i stort sett på samma tankesätt, om vi ser till följden från arbetsfallet. Visionsdokument är ännu viktigare när det gäller uppfyllandet av Use Cases samt hur de ska implementeras och hur de färdiga Use Cases ska knyts ihop. Detta eftersom anknytningspunkterna har ändrats i återanvändningsbara Use Cases i jämförelse med normala Use Cases. Samtidigt måste det klargöras att de får en viktigare status, så att analyser inte kan misstolkas, både för risktagarens behov samt för de problemen som ska lösas i projektet. Risken är annars att hela konceptet med användningen av återanvändningsbara Use Cases tappas bort.

5.5 Vikten av arkivering av Use Cases

En stor del med återanvändning är hur Use Cases hanteras mellan operationerna. Med andra ord hur de arkiveras. Telecas bild över en fungerande modell av återanvändning kräver ett nytt tänkande om hur de ska arkivera Use Cases så att de är lättillgängliga för andra projekt. Till själva arkiveringen är det viktigt att beakta att det kan vara en oberoende enhet som kanske inte är beroende av knappar att trycka på. Vilket löses i själva arkiveringen där de tar upp vilken styrning telefonen har. Namngivningen är av största vikt då.

Rubriken måste beskriva att det är ett Use Case och vilken del av projektet som det är del av, tillika det grafiska och att det inte är knapp styrt. Därefter kommer vilken revidering av Use Case det är och sist vad det berör (se bilaga Use Case image Teleca). Teleca visar att genom rätt rubricering är det möjligt att skapa ett fungerande arkiv för återanvändbara Use Cases. I ramverket lägger de på Teleca inte så stor vikt vid arkivering som de borde göra när de ska börja återanvända dem. I vanliga fall följer ett Use Case med projektet hela vägen och arkiveras med det. Vid återanvändning är det istället en förteckning över vilka Use Cases som används i projektet som behöver arkiveras i projektet. Use Cases som används går på detta sätt att ha lagrade i ett centralt arkiv för alla projekt på företaget.

5.6 Vinster vid användande av återanvändningsbara Use Cases

Vinster är av stor betydelse för att genomföra ändringar i modelleringen på ett företag. När det gäller Use Cases räknar Teleca med att det finns tidsvinster att göra, vilket medför såväl resultatekonomiska som personalekonomiska vinster. Resultatvinster bygger på att tiden kortas ner vilket i sin tur leder till att ett snabbare resultat uppnås. Personalen lägger mindre arbete på att skapa nya Use Cases som redan finns. Det är också möjligt att skicka ut modellerna tidigare till de olika berörda avdelningarna. Allt detta påverkar konkurrens effektiviteten genom ett snabbare genomlopp. Några konkreta siffror på detta finns ej än, utan är under utvärdering. Vi har endast kunnat konstatera att vårt antagande är det samma som de har på Teleca. Det vill säga de har också ansett, att

den förmodade vinsten de skulle göra på att återanvända Use Cases är, att de skulle kunna korta ner tiden det tar att skapa dem och därmed också minska sina utvecklingskostnader.

5.7 Summering av skillnader och likheter

För att lättare visa vad vi kommit fram till under vår undersökning har vi satt upp en tabell på delar inom Use Cases. Vi visar här skillnader och likheter mellan de olika typerna av Use Cases huvudfunktioner. Tabellen visar de skillnader som vi sett finns mellan hur Use Cases beskrivs i litteraturen och hur Use Cases på Teleca såg ut.

Tabell 5-1
Summering:

Beskrivning	Use Case	Återanvändningsbart Use Case	Kommentarer
Krav:	Villkor är definitionen av vad den klarar av eller vilka krav som systemet måste bevisa	Att klara av samma operation i olika projekt med samma krav som systemet måste bevisa	1)
Spårbarhet:	Att det ska vara möjligt att spåra en viss funktionalitet eller ett objekt genom ett helt projekt	Det ska vara möjligt att kunna finna rätt Use Cases till rätt operation genom spårbarheten i flera projekt	2)
Arkivering:	Use Cases lagras tillsammans med den ordinarie projekt dokumentationen för ett projekt	För projektet krävs en projektförteckning och ett centralt Use Case arkiv för revidering och lagring av Use Cases för alla projekt	3)
Icke funktionella krav:	För att skapa attribut beskrivningar som inte förklaras av systemets funktionella krav dokument	Här används inte några beskrivningar av de icke funktionella kraven i handlingsflödena	4)
Anknytningspunkt:	Use Cases har kopplings punkter till andra Use Cases	Huvudscenarion tjänstgör som kopplingar mellan Use cases	5)

Handlingsflöde/ Handlingsbasflöde:	Här sätts händelsen för Use Cases upp. Det förklaras hur det startas och avslutas.	Återanvändbara Use Cases läggs in i huvudscenariot. Men Use Cases förklarar inte eller nämner om var de börjar eller slutar.	6)
HuvudscENARIO:	Huvudscenario kan beskrivas som vad projektet ska uppnå.	Inriktningen är den samma i ett återanvändbart Use Case	7)

Kommentarer:

1) Vi har förstått att grundbegreppet för Use Cases är kravhantering. Syftet är att uppfylla eller att inte missa några krav. Krav är ett tillstånd eller en egenskap som ett system måste uppfylla eller inneha. För med återanvändbara Use Cases kan nya utvecklare känna sig något förvirrade, över tekniken att genomföra kravbeskrivningen på. Det är viktigt att det inte leder till att den generella funktionaliteten missas.

2) I Use Cases namnges varje projekt medan de i återanvändningsbara Use Cases namnges även för möjlighet att återanvändas. I någon annan form av spårbarhet skiljer sig inte de olika typerna av Use Cases.

3) Skillnaderna mellan de olika typerna av Use Case-teknikerna är att istället för att samla Use Cases i projektet har det i återanvändbara Use Case-projekt samlats referenser till Use Cases. Dessa referenser läggs i och tas från arkivet med färdiga Use Cases för att användas i projektet.

4) Vad som är speciellt med återanvändningsbara Use Cases är, att modelleraren kommer att påverka designmodellerna på grund av, att han blir manad att undvika navigeringsbara beskrivningar samt även undvika referat till användarens gränssnitt i flödet på Use Cases. Detta gör, att de ickefunktionella attribut, som skapas i vanliga Use Cases, inte kan finnas i dem som återanvänds, då där finns risk, att det skapar en unik händelse, vilket inte fungerar i återanvändbara Use Cases.

5) Om Use Cases ska återanvändas, kan där inte finnas anknytningspunkter, för detta knyter ihop till ett annat Use Case och då försvinner en del av återanvändbarheten med Use Cases. Detta är en stor skillnad mellan ett återanvändningsbart och ett vanligt Use Case, som binds till en anknytningspunkt, eftersom ett vanligt Use Case anses vara en unik händelse, medan det i den typ av utveckling mot moduler, som vi har undersökt, en händelse inte är unik utan används om och om igen

6) I ett handlingsflöde bör modelleraren förklara var scenariot börjar och var det slutar, om ett Use Case skall återanvändas visas det bara hur/varför det startar eller slutar. Där får inte finnas några navigeringsbara beskrivningar.

7) Skillnaden är störst i huvudscenariot även om utgångspunkten är densamma. I vanliga Use Cases så används huvudscenariot med de önskemål som kunden önskar. Av detta

skapas sub-Use Cases för de unika händelser som kan uppstå. Dokumentationen ligger separat från behandlingen av Use Cases. I återanvändbara Use Cases användes dokumentationen, vilket kommer att skrivas som ett grundläggande Use Case, som grund för att lägga in de Use Cases som behövs för att uppnå målen. Huvudscenariot kommer att fungera som den del där alla kopplingar finns mellan Use Cases och på det sättet vara det viktigaste Use Case som görs för det kommer aldrig att vara återanvändbart.

6 Slutsatser

Återanvändningsbara Use Cases har vissa delar som ser ut på samma sätt som i vanliga Use Cases men deras huvudfunktioner är i stort sett lika. För att kunna passa in i den här typen av företag är det vissa bitar som måste ändras, detta för att det ska bli en funktionell lösning för detta företag. Vi anser inte att det är en metod som är likvärdig med RUP. Det går förstås att diskutera om inte alla företag som använder Use Cases efter modellen RUP ändrar så att den ska passa in i deras verksamhet.

Telecas sätt skiljer emellertid sig så pass mycket och specifikt att vi vill påstå att det är ett mellanting mellan RUP och en helt ny teknik. Men ett bra ramverk bör gå att ändra för att passa in inom den bransch som använder det. Slutsatserna av vår undersökning blir därför att:

□ Det går att återanvända Use Cases men inte alla

Det är möjligt att återanvända Use Cases om de skrivs med återanvändning i åtanke från början. Det kan vara svårt att återanvända huvudflödena i en del Use Cases. Vinsten med att kunna återanvända stora delar av Use Cases gör ändå att vi anser att återanvändning kommer att gynna de företag som försöker att använda det.

□ Personalen måste utbildas i hur de ska använda återanvändningsbara Use Cases

Det är svårt att återanvända Use Cases om inte personalen utbildas i att skriva Use Cases som går att återanvända. Denna extra utbildning medför att det tar tid att införa återanvändningsbara Use Cases och kräver resurser som bör läggas till i den möjliga vinstberäkningen. Detta är dock enbart en initial kostnad tills personalen har anammat det nya arbetssättet.

□ Arkiveringen och spårbarheten är två faktorer som är komplicerade, men det är viktigt att de fungerar

Utan ett fungerande arkiv och en tydlig spårbarhet kommer de inte att hitta de Use Cases som behövs i andra projekt. Om det inte går att hitta dem inom projektet går det inte att hitta dem utanför projektet heller och det är ett måste för att återanvändbarhet ska gå att uppnå. Lösningen är att införa ett nytt sätt att arkivera sina Use Cases.

□ **Det behövs en standardiserad namnkonvention**

Detta för att alla ska kunna hitta de Use Cases de behöver i det projektet de jobbar med och även i de nya projekten där Use Cases ska återanvändas. Namnkonventionen är dessutom viktig eftersom det ska gå att stoppa in de återanvändningsbara Use Cases i nya projekt utan att spårbarheten i projektet försvinner eller förstörs.

6.1 Vidare forskning

Utvecklingen inom detta område är numera väldigt hett och vi ser att det inom flera områden används Use Cases. Vi har insett att både återanvändning och ommodellering av Use Cases kommer att öka för att göra vinster. Men ett problem är att dokumentationen kan bli betungande. Därför anser vi att det vore intressant att följa hur möjligheterna med återanvändning och ommodellering inom extreme programming kan fungera. Detta för att se om dokumentationen inom projekten kan minskas.

7 Referenslista

Böcker:

Backman, Jarl (1998) *Rapporter och uppsatser* Studentlitteratur, Lund.

Bergström Bo, Lundgren Pär, Flessa (1999) *Georg Grafisk kommunikation Medieboken*. Andra upplagan Kristianstads boktryckeri AB

Bittner, Kurt & Spence, Ian (2003) *Use Case Modeling* Pearson Professional Education, Harlow.

Bryman, Alan (2002) *Samhällsvetenskapliga metoder* Berlings Skogs, Trelleborg.

Cockburn, Alistair (2001) *Writing effective Use Cases* Third edition. Pearson Professional Education, Harlow

Deitel, Harvey & Deitel, Paul (2001) *C++ How to program Third edition* Prentice Hall, Upper Saddle River, New Jersey.

Engquist, Anders (2001) *Om konsten att samtala* Fälth & Hässler, Smedjebacken 2001.

Eriksson, Hans-Erik & Penker, Magnus & Lyons, Brian & Fado, David (2004) *UML 2 Toolkit* Wiley Publishing, INC, New York.

Guiney, Eamonn & Kulak, Daryl & Lavkulich, Erin (2003). *Use Cases: Requirements in Context* 2nd Edition Pearson Professional Education, Harlow.

Jacobson, Ivar & Griss, Martin & Jonsson, Patrik (1997). *Software Reuse: Architecture, Process and Organization for Business Success* Pearson Professional Education, Harlow.

Jacobson, Ivar & Christerson, Magnus & Jonsson, Patrik & Övergård Gunnar (1993) *Object-oriented software engineering: A Use Case driven approach* Pearson Professional Education, Harlow

Kruchten, Philippe (2003) *The rational unified process an introduction* Third edition. Pearson Professional Education, Harlow.

Patel, Runa & Davidson, Bo (1994) *Forskningsmetodikens grunder: att planera, genomföra och rapportera en undersökning* Studentlitteratur, Lund.

Preece, Jennifer & Rogers, Yvonne & Sharp, Helen (2002) *Interaction design: beyond human- computer interaction* John Wiley & Sons, Inc, Chichester.

Scott, Kendall (2001) *The Unified Process Explained* Pearson Professional Education, Harlow.

Skansholm, Jan (2002) *Java direkt med swing* Tredje upplagan Studentlitteratur, Lund

Kompendier:

Pettersson, Gertrud (1997) "Att skriva rapporter: Om formen och dess betydelse för innehållet." våren 1997

Interna källor:

Probasco, Leslee (2000) "The Ten Essentials of RUP The Essence of an Effective Development Process" Rational Software Corporation

Spence, Ian & Probasco Leslee (2000) "Traceability Strategies for Managing Requirements with Use Cases" Rational Software Corporation

Elektroniska källor:

Cockburn Alistair (Publicerad i STQE magazine, Mar/Apr 2002)

2006-07-04 *Use Cases, Ten Years Later*

<http://alistair.cockburn.us/crystal/articles/uctyl/usecasesentenyarslater.htm>

Comani Adriano 2006-05-24

Objektive view <http://www.iconixsw.com/ObjectiveView/ObjectiveView5.pdf>

Jacobson Ivar (Publicerad 2003) 2006-07-04

Use Cases: Yesterday, Today, and Tomorrow <http://www-128.ibm.com/developerworks/rational/library/775.html>

Pacestar Software 2006-07-10

Sample Use Case Diagram <http://www.pacestar.com/uml/sampleuc.htm>

Öhman Jan-Erik (Publicerad i BrainHeart Magazine) 2006-07-10

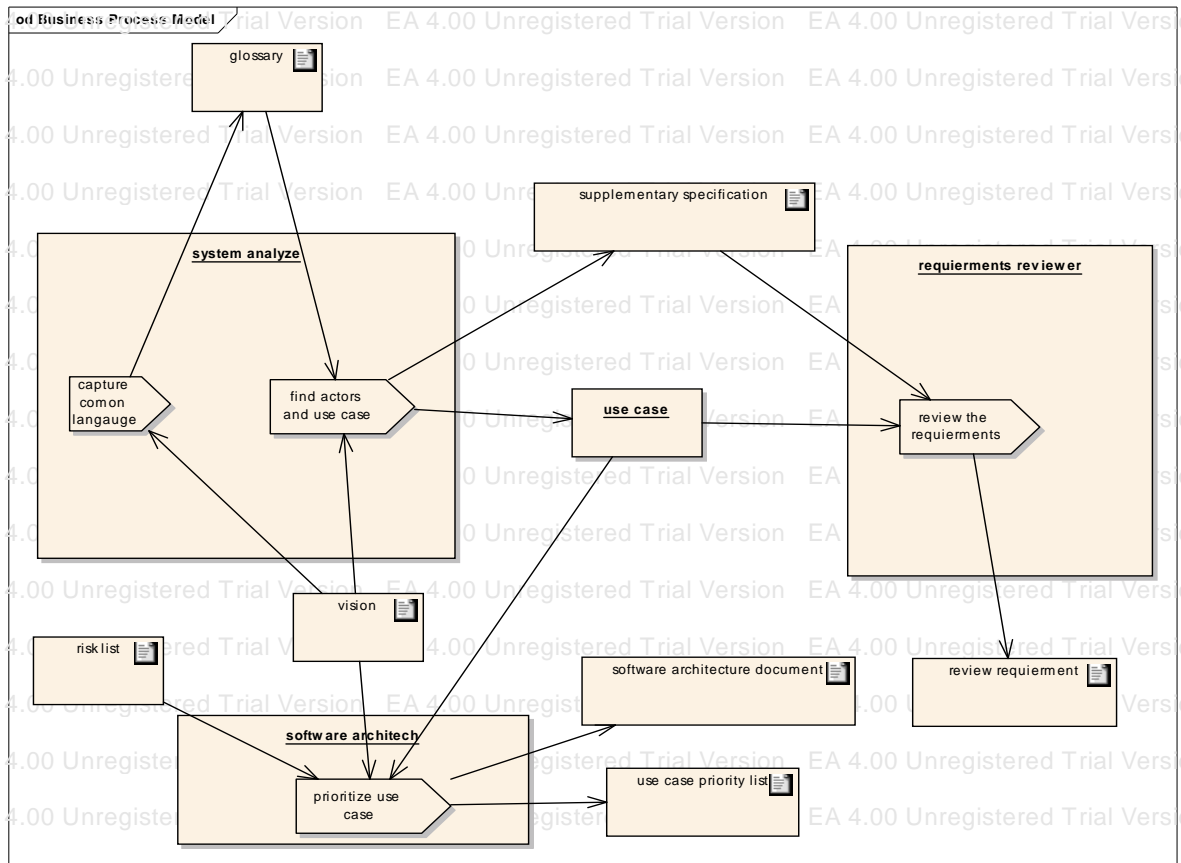
Face to face <http://dev.brainheart.com/images/uploads/issues/10/FaceToFace.pdf>

Bilagor

Akronymer och förkortningar

IBM	International Business Machines Corporation
UML	Unified modelling language
FURPS	Functionality, usability, reliability, performance and supportability
RUP	Rational Unified Process
CAS	Computer Aided Sales

Business Process Model



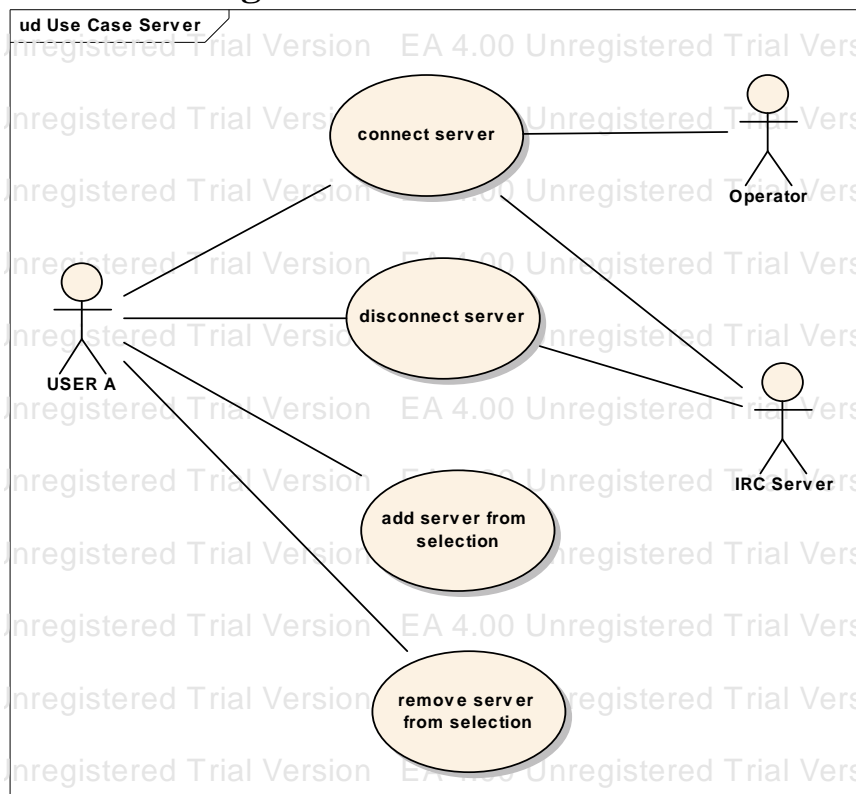
Reusable Use Case image Teleca

Ett exempel av ett återanvändningsbart Use Case som använts på Teleca

[RUC_2] Connect Server

Shows how to connect to a server

Use-Case Diagram



Flow Connect Server

User A wants to connect to a server

Basic Flow of Connect Server

Start of Use Case

User A wants to connect to a server.

<Scenario>

- User A fills in server name in the text area.
- User A selects connect

End of Use Case

User A has made a connection to a server.

Alternative Flow of Events in Connect Server

Alternative of Use Case Connect Server#1

User A wants to select a server from the server selection to make a connection

<Scenario>

- User A selects a server from the selections.
- Selected server name is automatically displayed.
- User A select connect

End of Use Case

User A has made a connection to a server from the server selection.

Alternative of Use Case Connect Server#2

User A types in a server name that doesn't exist

<Scenario>

- User A types in a server name.
- Server name doesn't exist.
- User A gets an error message
- User A Tries again.

End of Use Case

User A has made a connection to a server from the server selection.

Special Requirements

. None

Pre-conditions

User A must have done a login

Post-conditions

None

Extension Points

None

Relationships

Use Case relationship

- Actor User A.
- Actor Operator
- Actor server

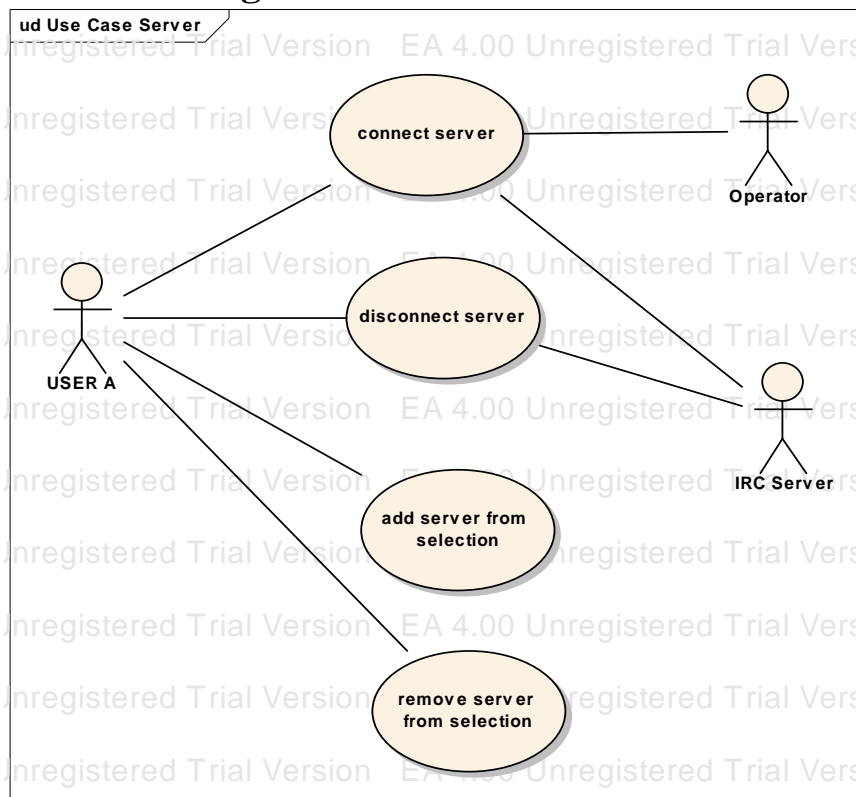
Use Case image Teleca

Ett exempel av Use Cases använda på Teleca

[UC_IRC_2] Connect IRC Server

Shows how to connect to a server

Use-Case Diagram



Flow Connect Server

User A wants to connect to a IRC server

Basic Flow of Connect Server

Start of Use Case

User A wants to connect to a IRC server.

<Scenario>

- User A fills in server name in the text area from the keyboard.

- User A presses the select button to connect

End of Use Case

User A has made a connection to a server.

Alternative Flow of Events in Connect Server

Alternative of Use Case Connect Server#1

User A wants to select a server from the server selection to make a connection

<Scenario>

- User A selects a server from the selection menu.
- Selected server name is automatically filled in the text area.
- User A press connect button

End of Use Case

User A has made a connection to a server from the server selection menu.

Alternative of Use Case Connect Server#2

User A types in a server name that doesn't exist

<Scenario>

- User A types in a server name in the text area.
- Server name doesn't exist.
- User A press connect button
- User A gets an error message on the display
- User A Tries again.

End of Use Case

User A has doesn't make a connection to a server from the server selection and have to select another server.

Alternative of Use Case Connect Server#2

User A types in a server name on the keyboard that doesn't exist

<Scenario>

- User A types in a server name in the text area.
- Server name doesn't exist.
- User A gets an error message on the display
- User A Tries again and succeed.

End of Use Case

User A has made a connection to a server from the server selection menu.

Special Requirements

. None

Pre-conditions

User A must have done a login

Post-conditions

None

Extension Points

User A comes from main menu and click chat button and gets to chat menu

Relationships

Use Case relationship

- Actor User A.
- Actor Operator
- Actor IRC server