

Forord	3
1 Introduksjon	5
2 Teori	7
2.1 Unified Process	7
2.1.1 Historie	8
2.2 Sentrale element ved UP	9
2.2.1 Unified Modeling Language	9
2.2.2 Use Case	10
2.2.3 Arkitektur	10
2.2.4 Iterativ og inkrementell	12
2.3 Unified Process sin livssyklus	13
2.4 Arbeidsflytar	15
2.4.1 Krav	15
2.4.2 Analyse	16
2.4.3 Design	17
2.4.4 Implementering	18
2.4.5 Test	19
2.5 Aktivitetsteori	21
2.5.1 Aktivitetssystemet	22
2.6 Læring	25
2.7 Empiriske systemutviklingsanalysar	27
2.7.1 Kvalitative studium av systemutvikling	27
2.7.2 Aktivitetsteori i studium av systemutvikling	28
2.7.3 Kulturell kontekst	30
2.7.4 Tolkande casestudium	31
2.7.5 Metoden si rolle	32
3 Forskingsspørsmål	35
4 Forskingsdesign	37
4.1 Datainnsamling	39
4.1.1 Observasjon og uformelle samtalar	39
4.1.2 Dokumentanalyse	40
4.1.3 Intervju	41
4.1.4 Transkribering	42
4.1.5 Dataanalyse	44
4.2 Forskingsetikk	45
5 Evaluering av gjennomføring	47
5.1 Observasjon og uformelle samtalar	47
5.2 Dokumentanalyse	47
5.3 Intervjuguide	48
5.4 Gjennomføring av intervju	48
5.4.1 Første intervju	49
5.4.2 Andre intervju	49
5.4.3 Tredje intervju	50
5.4.4 Fjerde intervju	51
5.4.5 Femte intervju	52
5.4.6 Pilskog sine intervju.	52
5.5 Formatering og transkribering av intervju	53

5.6 Første gjennomlesing, analysestart	53
6 Analyse	55
6.1 Utviklingsprosessen som eit aktivitetssystem	55
6.1.1 Subjekt	55
6.1.2 Objekt	56
6.1.3 Verktøy	56
6.1.4 Samfunn	58
6.1.5 Reglar	59
6.1.6 Arbeidsdeling	60
6.1.7 Aktivitetsnivå	61
7 Funn og observasjonar	63
7.1 BU og UP generelt	64
7.1.1 Tilpasse metoden til prosjektet	65
7.1.2 Iterasjonar	67
7.1.3 Arbeidsflytar	68
7.2 RUP verktøy eller regel	70
7.3 Kommunikasjon og dokumentasjon	72
7.4 Læringseffektar	76
7.4.1 Domenekunnskap	77
7.4.2 Metodekunnskap	79
7.4.3 Teknologisk kunnskap	80
7.5 Oppsummering	83
8 Konklusjon og oppsummering	87
8.1 Kva kunne vore gjort annleis	88
8.2 Vidare arbeid	89
Referansar	91
Web referansar	95
Figurliste	97
Vedlegg 1: Generell intervjuguide	99

Forord

Dette arbeidet, *“RUP i praksis – Ein studie av kommunikasjon og læringseffektar”*, markerer avslutninga på utdanninga mi ved Universitetet i Bergen.

Eg vart teken opp som hovudfagsstudent ved institutt for informasjonsvitskap hausten 2001 og har fullført følgjande 5 vektta hovudfagskurs:

- IV314 ”Gaming: methodology & practice” (haust 2001)
- IV351 ”Forskningsmetodologiar innan pedagogisk informasjonsvitskap” (haust 2001)
- IV352 ”Samtidige og historiske perspektiv på pedagogisk informasjonsteknologi (vår 2002)
- IV300 ”Informasjonsvitskapeleg forskning og forskingsmetode” (vår 2002)

I tillegg har eg ved Universitetet i Bergen fullført mellomfag i Medievitskap og grunnfag i Administrasjon og Organisasjonsvitskap.

Med bakgrunn i utdanninga mi her ved Institutt for Informasjonsvitskap har eg gjennom fleire semester arbeidd teoretisk med systemutviklingsprosessar. Dette prosjektet gjev meg ein sjanse til å studere ein slik utviklingsprosess, og sjå korleis det fungerer i røynda. I tillegg har eg tileigna meg fagleg kunnskap innan pedagogisk informasjonsvitskap, eg finn det difor interessant å studere kva for læringseffektar som kan finnast i ei gruppe som driv med systemutvikling, og om eg der kan dra veksle på den teoretiske kunnskapen min om samarbeidslæring. Endeleg håper eg gjennom dette arbeidet å stille betre budd til å møte arbeidsmarknaden, etter å ha vore inne og studert korleis eit verkeleg prosjekt kan skride fram.

Eg vil her takke medstudent Pilskog og rettleiar Bjørnar Tessem ved Institutt for informasjonsvitskap som har vore til stor hjelp gjennom heile dette prosjektet. I tillegg vil eg rette ei stor takk til dei tilsette og mine respondentar ved Bjørgvin

Utvikling, som gjennom heile prosessen har vore hjelpsame og positive til arbeidet mitt.

1 Introduksjon

Systemutvikling har gjennom dei seinare åra gått frå å vere ein aktivitet som vert utført av enkeltpersonar eller mindre grupper, til å vere ein aktivitet som sysselset bedrifter av varierende storleik. Krav til stadig meir komplekse, effektive, og driftssikre system har skapt behov for å arbeide etter faste prosedyrar og reglar, med andre ord, etter ein vedteken metodologi.

Fleire ulike metodologiar har vorte nytta gjennom dei siste åra med vekslende hell. Rational Unified Process, fell inn under paraplyen objektorientert utvikling, og har som mål å gjere utviklinga rimelegare, tryggare og grundigare. I dette arbeidet vil analysere og vurdere denne metoden i bruk.

Dette arbeidet er del av eit større vitenskapleg prosjekt ved Institutt for Informasjonsvitenskap i Bergen. Prosjektet retta mot empiriske studium av ulike systemutviklingsmetodar.

I oppgåva mi vil eg studere og analysere metoden Rational Unified Process slik firmaet Bjørgvin Utvikling nyttar den i sitt prosjekt Forv@lting. Oppdragsgivar for det konkrete prosjektet er eit statleg departement, og systemet under utvikling skal nyttast i den offentlege forvaltninga. Bedrifta, prosjektet, oppdragsgivar og prosjektgruppa medlemmar har alle vorte gjort anonyme etter avtale med bedrifta.

Bjørgvin Utvikling (BU) er eit heileigd statleg aksjeselskap. Utviklingsavdelinga til BU opererte tidlegare i eit sjølvstendig konsultantselskap, men etter lengre tids samarbeid vart dei tekne opp i det som i dag er BU. Utviklingsavdelinga arbeider mykje med drifting av eksisterande system, men også med utvikling.

Forv@lting er eit stort prosjekt som skal effektivisere og kvalitetssikre eit spesifikt segment av den offentlege forvaltninga. Oppdragsgjevar er eit departement. Prosjektet er relativt stort og har som eit av måla sine å legge til rette for bruk av elektroniske dokument. Systemet skal ved ferdigstilling køyre på ein vevplattform, og er såleis delt i ein klient- og ein tenardel

Rational Unified Process (RUP) er ein velkjend prosess for objektorientert systemutvikling. Metoden er ein variant av Unified Software Development Process og vert distribuert av Rational Software Company. Formålet med prosessen er å sikre at utviklinga av komplekse system skjer med låg risiko på ein grundig og rimeleg måte. (Rational.com)

I arbeidet med å analysere BU sin bruk av Forvaltning vil eg nytte Aktivitetsteori som rammeverk. Aktivitetsteori nyttar grovt sagt aktiviteten som analyseeining og har ei form som gjer den passende til studium der menneskeleg aktivitet, og ikkje minst samspel mellom menneske, vil stå i fokus.

Den vidare organiseringa av denne oppgåva vil vere som følgjer. Først vil eg ta for meg dei grunnleggande elementa ved RUP som metode, og så kva som kjenneteiknar den. Eg vil etter dette presentere det analytiske rammeverket mitt – Aktivitetsteori. Eg vil også drøfte korleis dette er relevant for læring. Til slutt i dette teorikapittelet vil eg presentere ulike empiriske studium av systemutviklingsprosessar, som eg vil bygge på i mitt eige arbeid.

I kapittel tre vil eg presentere mitt eige forskingsspørsmål og kva eg vil legge særleg vekt på i forskinga. Etter dette vil eg presentere forskingsdesignet mitt. Ein presentasjon og ei evaluering av gjennomføringa vil følgje i neste kapittel, før analysefasen tek til i kapittel seks. Mine funn og observasjonar vert presentert i kapittel sju. I siste og avsluttande kapittel vil eg samle trådane og summere opp mine funn i dette arbeidet. Eg vil og presentere ulike skisser for vidare arbeid innan fagområdet.

2 Teori

Björgvin Utvikling (BU) nyttar i systemutviklinga etter eige utsegn ein modifisert versjon av Rational Unified Process (RUP). Utgangspunktet for omtalen av RUP i denne oppgåva vil i hovudsak basere seg på Jacobsen, Booch og Rumbaugh si bok "The Unified Software Development Process".

Aktivitetsteori er rammeverket verksemda og systemutviklinga til BU skal analyserast og tolkast ut ifrå. Aktivitetsteori er, som det vil gå fram av presentasjonen under, eit godt rammeverk for å vurdere BU si systemutvikling

Denne forskinga skal i hovudsak sjå på systemutviklinga som ein læringsprosess, med fokus korleis læring og kommunikasjon vert endra gjennom å nytte RUP som systemutviklingsmetode. Eg vil presentere definisjonar og læringssyn som er relevante for forskingsspørsmålet.

2.1 Unified Process

Unified Process (UP) er i følgje skaparane eit svar på kravet om stadig større behov for ein kontrollert arbeidsprosess, grunna stadig meir komplekse og større system. UP er i første rekke ein prosess for programvareutvikling. UP er komponentbasert, noko som tyder at programvaren vert bygd opp av programvarekomponentar tett knytt opp mot eit vel definert grensesnitt.

Omgrepa Rational Unified Process (RUP) og Unified Process vert ofte nytta om kvarandre. Det må her presiserast at når ein snakkar om RUP, så er det snakk om ein detaljert og kommersiell førekomst av UP utvikla og distribuert av Rational Software Company. Rational presenter RUP som:

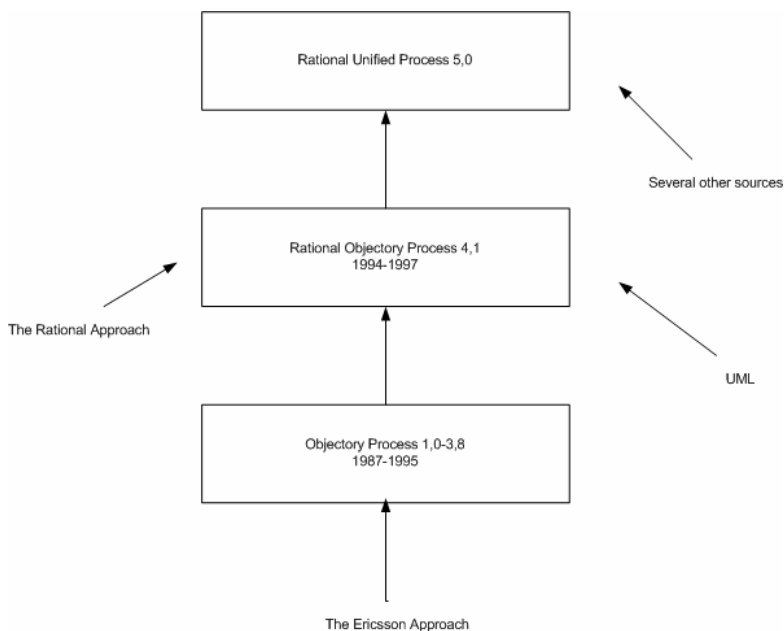
"The Rational Unified Process, or RUP, is a web-enabled set of software engineering best practices that provide you with guidance to streamline your team's development activities."

(Rational.com)

UP slik Jacobsen, Booch og Rambaoug presenterer det i boka *The Unified Software development process*, er det teoretiske fundamentet, ein rein ikkje-kommersiell presentasjon av korleis dei meiner ein systemutviklingsprosess best skal finne stad. Nemninga Unified Process vil i denne teoridelen verte nytta framfor RUP ettersom det er dei teoretiske tankane som ligg til grunn for systemutviklingsprosessen meir enn det kommersielle produktet Rational leverer, som vert presentert.

2.1.1 Historie

Unified Process er resultatet av over tretti år med utvikling og praktisk bruk. Unified Process si utvikling som produkt følgjer utviklingsrekka som vist i figur 1 under. Med utgangspunkt *The Ericsson Approach* vart *the Objectory Process* presentert for første gang i 1987. *The Objectory process* vart vidareutvikla gjennom fleire versjonar fram til 1996. Ved innføring av UML og *the Rational Approach* vart den døypt om til *Rational Objectory Process 4.1*. Eit par år seinare - i 1998, etter fleire endringar og tillegg, presenterte Rational *the Rational Unified Process 5.0*.



Figur 1 Rational Unified Process si historiske utvikling (Jacobsen et al. 1999).

2.2 Sentrale element ved UP

Eg vil under dette punktet presentere dei sentrale element som utgjer grunnelementa for Unified Process sine systemutviklingstankar.

2.2.1 Unified Modeling Language

Unified Modeling Language (UML) vart utvikla av Jacobsen, Booch og Rumbaugh på ei tid der behovet for eit presist og uniformt visuelt språk var tydelig. Etter nokre tidlege versjonar vart versjon 1.1 publisert i november 1997. UML har i mange sine auge, meir eller mindre vorte standard språk for modellering i objektorienterte systemutviklingsprosjekt. UP nyttar UML til utarbeiding av modellar av programvaren under utvikling. Slik sett er UML ein integrert del av Unified Process.

“The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing and documenting the artifacts of software systems. It simplifies the complex process of software design, making a “blueprint” for construction.”
(Rational.com)

UML definerer tolv typar diagram, delt inn i tre kategoriar. Fire diagramtypar representerer den statiske applikasjonsstrukturen; fem representerer forskjellige aspekt av dynamisk oppførsel og tre representerer måtar å organisere og handsame ulike modular ved applikasjonen. (www.omg.org)

- Strukturdiagram: klassediagram, objektdiagram, komponentdiagram, og utrullingsdiagram.
- Oppførselsdiagram: Use Case diagram, sekvensdiagram, aktivitetsdiagram, samarbeidsdiagram og tilstandsdiagram.
- Modelladministreringsdiagram: pakkar, undersystem og modellar

2.2.2 Use Case

“A use case specifies a sequence of actions, including variants that the system can perform and that yield an observable result of value to a particular actor.”

(Jacobsen et al. 1999:41)

Eit use case skal fange opp dei funksjonelle krava til eit system. Samla vil dei ulike use case utgjere ein use case-modell som skal beskrive den komplette funksjonaliteten til eit systemet. Use case vert utarbeidd ved å stille seg spørsmålet: Kva skal systemet gjere for kvar brukar? Med brukar forstå ein her ikkje berre personar som brukar systemet, men og alle andre komponentar som på ulike måtar er knytt opp mot systemet under utvikling.

Use case styrer den vidare progresjonen i form av design, implementering og testing av systemet. Basert på use case-modellen, utarbeider utviklarane ei rekke design- og implementeringsmodellar som skal realisere use case. For kvar etterfølgjande modell vert det sjekka at dei er i samsvar med use case-modellen. På denne måten bind use case utviklingsprosessen saman, så vel som å initiere den. Use case driven tyder at utviklingsprosessen følgjer ein flyt, den skrid fram gjennom ei rekke arbeidsflytar som alle tek utgangspunkt i use case. Use case vert spesifisert, designa og er til slutt kjelda som testarane utviklar test casea sine frå.

Use case vert ikkje utvikla i isolasjon. Use case vert utvikla i tandem med systemets arkitektur. Use casea driv systemets arkitektur og systemets arkitektur verkar inn på val gjort i use casea. Dette resulterer i at arkitektur og use case modnast gjennom systemets livssyklus.

2.2.3 Arkitektur

” The software architecture concept embodies the most significant static and dynamic aspects of the system” (Jacobsen et al 1999: 6)

Arkitekturen vert utvikla i høve til use casea. Kvar programvare treng både funksjon og utforming. Berre den eine eller den andre vil ikkje vere tilstrekkeleg. Dei to må

vere godt balanserte for å oppnå eit godt produkt. Skaparane omtalar det som den klassiske ”høna og egget” - problematikken:

”It is a ”chicken and egg” problem. On the one hand, the use case must, when realized, fit in the architecture. On the other hand, the architecture must leave room for realizations of all required use cases, now and future. In reality, both the architecture and the use cases must evolve in parallel” (Jacobsen et al. 1999:6)

Arkitekturen er nyttig for å presentere:

- Systemforståing
- Organisering av utviklinga
- Fostre gjenbruk
- Sjølve utarbeidinga av systemet

Når ein organisasjon skal utvikle eit system, er det nødvendig at systemet er forståeleg for alle involverte partar. Det å gjere moderne system forståeleg er ei stor utfordring av fleire grunnar:

- Dei inneheld kompleks oppførsel.
- Dei opererer i komplekse miljø.
- Dei er teknologisk komplekse.
- Ofte kombinerer dei distribuerte berekningar, kommersielle produkt og plattformer, og i tillegg gjenbruk av tidlegare komponentar og rammeverk.
- Dei skal tilfredsstille behov for individ så vel som for organisasjonar.
- I enkelte tilfelle vil og systema vere så store at utviklinga er geografisk spreidd, noko som aukar problema ved å koordinere utviklinga.

Desse faktorane, som er i konstant endring, summerer i følgje skaparane (Jacobsen et al. 1999) opp dei potensielle vanskanne ein kan ha med å forstå kvarandre. Det å gjere utviklinga arkitektursentrert, er ein måte å forhindre misforståingar aktørane i mellom.

I forhold til organiseringa av utviklinga meiner skaparane (Jacobsen et al. 1999) at ved å dele opp systemet i mindre undersystem med klart definerte grensesnitt og ved å tileigne grupper eller individ ansvar for dei ulike undersystema, kan arkitekturen redusere kommunikasjonsbehovet mellom gruppene som arbeider med dei ulike undersystema. Eit stabilt grensesnitt gjer at arbeidet på fleire felt kan skride fram samstundes, og gruppene kan arbeide uavhengig av kvarandre.

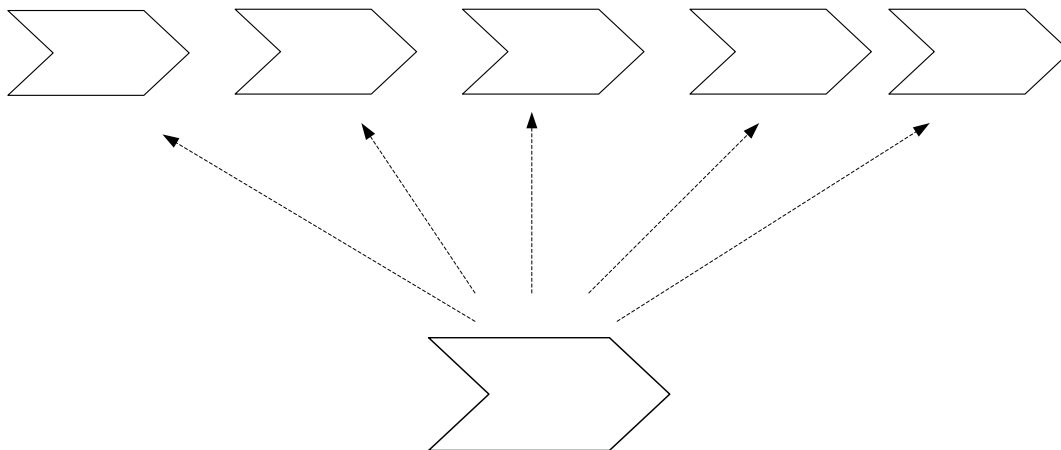
Programvareindustrien har enno ein lang veg å gå når det gjeld standardisering av komponentar og arkitektur (Jacobsen et al. 1999). God arkitektur og eksplisitt grensesnitt er steg i rett retning. Ein god arkitektur gir utviklarane gode rammeverk å arbeide ut ifrå. Gjennom eit godt rammeverk kan ein oppnå ein rasjonaliseringsvinst i gjenbruk av undersystem. Utviklarane av UML (Jacobsen et al. 1999) trur at språket vil framskande ein meir komponentbasert prosess. Dei meiner at eit standard modelleringspråk er ein føresetnad for å bygge domenespesifikke komponentar som kan nyttast fleire gonger.

Eit kvart system av ein viss storleik vil vere avhengig av å kunne utvikle og endre seg. Slike endringar vil finne stad både under sjølve systemutviklinga og ikkje minst når systemet vert teke i bruk. For at systemet skal takle endingar og vere open for ny funksjonalitet, er ein avhengig av ein god og presis arkitektur. System med dårleg definert arkitektur vil forvitte over tid og svekkast gjennom nye oppdateringar (patches), i høve til å drive kostnadseffektivt.

2.2.4 Iterativ og inkrementell

Arbeid med iterasjonar består veldig forenkla av å dele eit prosjekt opp i fleire mindre prosjekt. Kvar iterasjon inneheld alt eit "vanleg" utviklingsprosjekt inneheld: utarbeiding av planar, gjennomarbeiding av ei rekke arbeidsflytar (krav, analyse og design, implementering og testing) og publisering. Iterasjonane kan ikkje sjåast på som sjølvstendige einingar. Dei er stadium i ein prosess, og er merka av det. Kvar iterasjon kan og kallast eit minifossefallsprosjekt, fordi den går igjennom alle dei klassiske fasane i eit slikt prosjekt.

Kvar iterasjon består av fem arbeidsflytar, som vist i figur 2. I kvar iterasjon vil utviklarane identifisere og spesifisere dei relevante use casea, lage eit design med den valde arkitekturen som rettesnor, implementere designet i komponentar, samt verifisere at komponentane tilfredsstillir krava i use casea.



Figur 2 Arbeidsflytar i ein iterasjon (Jacobsen et al. 1999)

Dei første iterasjonane er i hovudsak retta mot å definere prosjektets omfang, fjerne kritiske risikofaktorar og utarbeide ein grov arkitektur. Etersom prosjektet skrid fram og ein gradvis reduserer talet på risikofaktorar og implementerer komponentar, endrar forma på iterasjonane seg. Kvar iterasjon resulterer no i at ein produserer inkrement, som vert ein tilvekst til det endelege produktet. Desse inkrementa kan visast til kundar og brukarar, og dei kan igjen gi verdfull

Requirements

2.3 Unified Process sin livssyklus

UP repeterer seg sjølv gjennom ein serie av syklusar, som igjen utgjer "livet" til eit system. Kvar syklus består av fire fasar: start (inception), utdjujing (elaboration), konstruksjon (construction), overføring (transition).

Startfasen består i finne svaret på kva systemet under utvikling skal kunne gjere for brukaren, korleis skal arkitekturen vere, korleis prosjektet skal skride fram og kva vil det koste. Ein begynner å modellere use case for å svare på det første spørsmålet. Arkitekturen er berre vagt definert, den vil i grove trekk berre omfatte dei mest

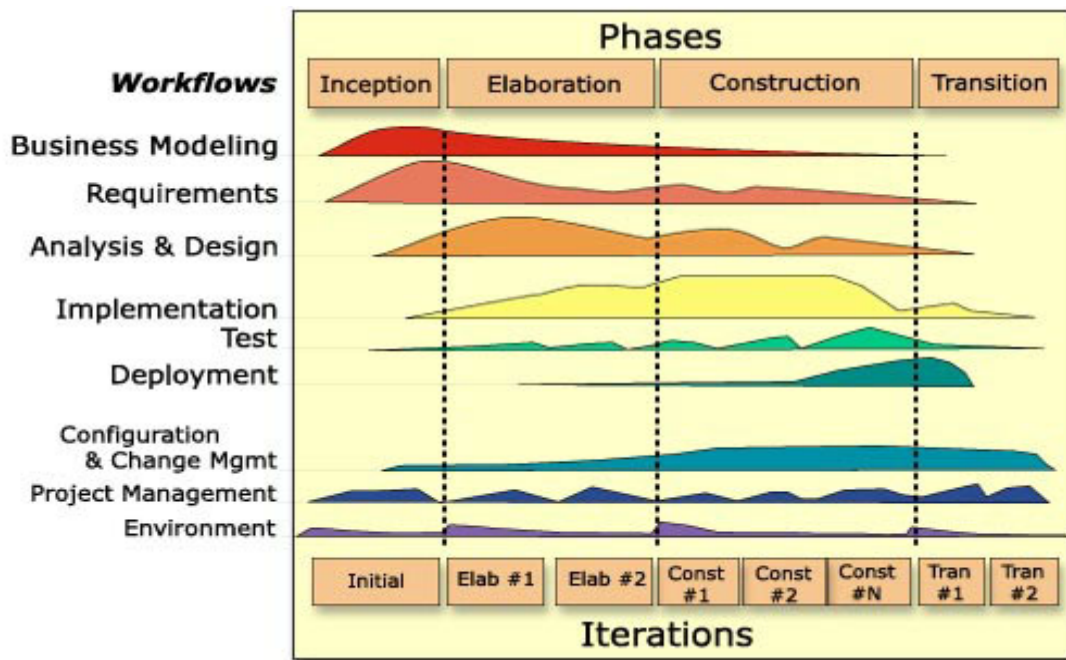
sentrale undersystema. Vidare vil ein identifisere og prioritere dei mest sentrale risikofaktorane for prosjektet. Ein legg ein plan for utdjupingsfasen og freistar å estimere kostnader og tidsrammer for heile prosjektet.

I utdjupingsfasen vert use casea og arkitekturen spesifisert i detalj. Forholdet mellom arkitekturen og resten av systemet står i høgsetet. Før fasen kan avsluttast må utviklarane vere sikre på at use casea, arkitekturen og vidare planer er stabile nok, og at ein har kontroll over dei ulike risikofaktorane.

I konstruksjonsfasen vert produktet utvikla. Programvara vert bygd på og rundt arkitekturen som er utvikla tidligare med utgangspunkt i use casea. Mot slutten av denne fasen skal alle vedtekte use case vere realiserte, og ein skal ha eit produkt som fungerer rimeleg feilfritt.

Overgangsfasen dekker perioden der produktet nærmar seg ferdigstilling og gjerne vert presentert i ein betaversjon. Betaversjonen vert testa av erfarne brukarar som rapporterer feil og manglar tilbake til utviklarane. Overgangsfasen inkluderer også levering av endeleg produkt, samt opplæring, feilretting og vedlikehald. Kvar av desse fasane er igjen delt opp i iterasjonar, som omtalt over. Kvar syklus resulterer i ei publisering av systemet, og ved kvar publisering vil der vere eit produkt klar for overlevering. Dette produktet består i køyrelege og kompilerlege komponentar som inneheld kjeldekode, pluss manualar og tilhøyrande resultat.

I dei ulike fasane finn vi og dei ulike arbeidsflytane: krav, analyse, design, implementering og testing. Korleis fasane og arbeidsflytane går inn i kvarandre og fordeler seg i dei ulike iterasjonane er vist i figur 3.



Figur 3 Unified Process sin livssyklus (Rational.com)

2.4 Arbeidsflytar

2.4.1 Krav

Målet med kravinnsamlinga er å styre prosjektet i rett retning. Dette målet oppnår ein ved å beskrive systemkrava godt nok, slik at kunde og utviklar kan verte einige om kva systemet skal og ikkje skal gjere. Ei stor utfordring i kravarbeid er at kundane oftast ikkje sjølve er dataekspertar. Utviklarane må difor uttrykke krav på ein måte som både er detaljert nok og godt forståeleg for kunden. Resultatet av kravinnsamlinga skal og vere til hjelp for prosjektleieren når han skal planlegge iterasjonane og publiseringa. Der er i hovudsak to typar krav som skal fangast opp, funksjonelle og ikkje- funksjonelle krav.

I kravarbeidet er det nødvendig å ha oversikt over konteksten systemet utviklast i. Det vert foreslått to måtar å gjere dette på (Jacobsen et al. 1999); utvikling av ein domenemodell eller utvikling av ein forretningsmodell (businessmodell) Ein domenemodell er eigentleg ein spesiell førekomst av ein komplett forretningsmodell. Slik sett er det å utvikle ein forretningsmodell eit betre alternativ enn å utvikle ein domenemodell.

Forretningsmodellering er ein teknikk for å forstå dei ein-skilde forretningsprosessane i ein organisasjon. Forretningsmodellering er støtta av to typar UML-modellar: use case og objektmodellar. Begge er definert i den forretnings-spesifikke utvidinga av UML. Det å utvikle slike modellar består i å identifisere arbeidarane (workers) og artefakta som tek del i kvar ein-skild arbeidsflyt. Ein arbeidar representerer ein posisjon som kan verte tileigna ein person eller eit team, og den spesifiserer kva ansvar og moglegheiter arbeidaren har.

Å samle krava ved hjelp av use case er i følgje utviklarane (Jacobsen et al. 1999) ein systematisk og intuitiv måte å arbeide på, ein legg særleg fokus på verdien kvar del har for den ein-skilde brukar. Ved å anvende use case vert ein tvungen til å tenke på kven brukarane er, og kva kunde- eller oppdragsbehov som kan realiserast gjennom use casea? Jacobsen hevdar vidare at funksjonelle krav vert naturleg strukturerte i use case og sidan dei fleste ikkje-funksjonelle krava er spesifikt knytt til eit use case vert også desse fanga opp gjennom use case-modelleringa. Dei resterande ikkje-funksjonelle krava, som er felles for fleire use case, vert nedteikna og handsama separat i eit dokument ein kallar supplementerande krav.

2.4.2 Analyse

I analysefasen analyserer ein krava gjennom å raffinere og strukturere dei betre. Formålet med dette arbeidet er å få ei meir presis forståing av krava og å utarbeide ei skildring av krava som er lett å vedlikehalde, og som hjelper til å strukturere heile systemet inkludert arkitekturen.

I analysefasen vert det nytta ein konseptuell objektmodell kalla analysemodell. Modellen yter ein meir presis spesifikasjon av krava enn resultata frå kravinnsamlinga, inkludert use case-modellen. Analysemodellen har meir uttrykksfull kraft og formelle uttrykk slik som interaksjonsdiagram, som vert nytta til å beskrive dynamikken i eit system. Modellen har og ein struktur som fokuserer på evne til vedlikehald, og skal ta høgde for feiltoleranse innan kravhandsaming og gjenbruk. Ein analysemodell er laga i eit språk retta mot utviklarane. Den kan difor vere meir formell og nyttast til å resonnerer rundt indre forhold i systemet. Dette i motsetning til modellar i kravdelen som og skal vere forståelege for kunden.

Analysemodellen strukturerer krava på ein måte som gjer dei lettare å forstå, klargjere, endre og halde ved like. Analysemodellen er det første steg i retning av ein designmodell.

Av konkrete element som analysemodellen inneheld kan nemnast: analysepakkar og servicepakkar, og tilhøyrande avhengigheit og innhald. Analysepakkane kan lokalisere endringar i ein forretningsprosess, aktørane sin oppførsel, eller eit sett tett knytte use case. Servicepakkane vil lokalisere endringar på einskilde tenester systemet yter, og dei spesifiserer eit elementært instrument for å fostre gjenbruk gjennom analysen.

Det vert og utarbeidd analyseklasser med tilhøyrande ansvarsområde, attributt, forhold og spesielle krav. Kvar av kontroll-, entitet- og grense (boundary)- klassene vil lokalisere endring i høve til den vanlege oppførselen og informasjonen dei representerer. Use case-realiseringa, som beskriv korleis use casea vert raffinert i forhold til samarbeid internt i analysemodellen og use casea sine spesielle krav. Use case-realisering vil lokalisere endringar i use casea, sidan endring i eit use case kan medføre endring i realiseringa. Det arkitektoniske synet i analysemodellen inkluderer dei mest signifikante elementa, og vil halde styr på endringar i arkitekturen. Det arkitektoniske synet vert nytta når designmodellen vert utarbeidd.

2.4.3 Design

I designfasen skal systemet finne si form, inkludert arkitekturen, på ein slik måte at det lever opp til alle krav, funksjonelle og ikkje funksjonelle. Designarbeidet tek utgangspunkt i analysemodellen som pålegg designen ein struktur ein skal legge vekt på å behalde gjennom utforminga av systemet.

Eit viktig formål med designarbeidet er å utvikle ei djup forståing av ikkje-funksjonelle krav og avgrensingar relatert til programmeringsspråk, gjenbruk av komponentar, operativsystem, distribusjon og fleirbruksteknologi (concurrency),

databaseteknologi, brukargrensesnitt-teknologi, transaksjonshandterings- teknologi etc.

Designen skal legge til rette for implementeringsaktivitetane ved å fange opp krava til dei individuelle undersystema, grensesnitt og klasser. Det skal og vere i stand til å dekomponere implementeringsarbeidet til meir handterlege bitar som vert fordelt på ulike utviklings team, gjerne samtidig. Designen skal og fange kontakta mellom dei ulike undersystema i ei tidleg fase av utviklinga. Det er og viktig å nytte ein felles notasjon for å kunne kommunisere og visualisere designen på ein måte som alle forstår. Designen vil vere som eit skjelett der implementeringa består i å få kjøtt på beina utan å endre strukturen.

Hovudresultatet av designprosessen vert ein designmodell som skal vere i samsvar med strukturen som vert sett fram i analysemodellen. Designmodellen skal virke som ein blåkopi av den føreståande implementeringa. Modellen er også ei hierarkisk framstilling av dei ulike undersystema og deira koplingar og innhald. Designklassar, aktive klassar inkludert operasjonar, attributtforhold og implementeringskrav. Generelt er designklassene utleidd frå analyseklassene. Use case-realiserings finn og stad i designmodellen, som og er basert på arbeidet gjort i analysedelen.

Designarbeidet resulterer og i ein utrullingsmodell (deployment) som skildrar nettverkskonfigurasjonane systemet skal distribuerast på. Denne modellen inneheld dei ulike nodane og deira eigenskapar og bindingar, ei innleiande avbilding av aktive klasser på nodane, og eit arkitektonisk syn på modellen som inkluderer dei viktigaste arkitektoniske elementa

2.4.4 Implementering

Implementeringsfasen tek utgangspunkt i resultatata frå designet og realiserer systemet i form av komponentar som kjeldekode, script, køyrelege element og liknande. Det meste av arkitekturen er allereie på plass gjennom fasane til og med design, så i implementeringsfasen skal ein gje substans til systemet som heile.

Det er sentralt i tankegangen til UP (Jacobsen et al. 1999) å utvikle ein plan for systemintegreringa for kvar iterasjon. Tilnærminga er som tidlegare nemnt inkrementell, dette medfører at systemet skal implementerast i etterfølgjande handterlege steg.

Systemet vert distribuert gjennom å knytte til køyrelege element på nodane i utrullingsmodellen. Denne distribueringa er i hovudsak basert på dei aktive klassene ein har funne i designarbeidet. Vidare skal designklassene og undersystema i designfasen også implementerast, med tilhøyrande avhengigheit, grensesnitt og innhald.

Designklassene vert i særleg grad implementert som filkomponentar som inneheld kjeldekoder. Det vert og gjennomført einingstestar (Unit test) av komponentar før dei vert integrert gjennom å compilere dei og linke dei saman til ein eller fleire kjørelege einingar. Dei vert deretter sendt til integrering og system testing.

Implementeringa resulterer og i ei raffinering av det arkitektoniske synet på utrullingsmodellen, når køyrelege komponentar vert tilknytta på nodane. Kvart realiserte tillegg i implementeringsfasen vert integrasjonstesta og kan og verte systemtesta i testfasen

2.4.5 Test

I denne fasen er hovudarbeidsoppgåva å verifisere alle produkta frå implementerings-fasen. Dette arbeidet vert utført gjennom å teste kvar einaste tilvekst (build). Systemet som heile skal gjennom fleire testar før det vert overlevert eksterne partar.

Som i dei andre fasen vert det utvikla ein modell, ein testmodell, som her skal innehalde testcase, prosedyrar og komponentar. Integrasjonstest skal gjennomførast for kvar einaste tilvekst i iterasjonen og systemtest skal gjennomførast ved slutten av kvar iterasjon. Det skal designast og implementerast testcase som spesifiserer kva som skal testast, skape testprosedyrar som spesifiserer korleis testane skal

gjennomførast, og det skal lagast køyrelege testar for automatiske testar der det er mogleg. Systemtilvekstar som ikkje passerer testane vert send tilbake til dei respektive arbeidsflytane, eksempelvis design eller implementering, slik at feila kan rettast opp.

2.5 Aktivitetsteori

Aktivitetsteori (AT) vil vere rammeverk for dette studiet av systemutviklingsprosessar i BU. Aktivitetsteori er eit godt rammeverk å nytte når ein skal fokusere på dei menneskelege og kulturelle aspekta ved ein systemutviklingsprosess. Nemninga aktivitetsteori kan vere noko misvisande. Aktivitetsteori er ikkje ein teori i streng tyding, som kan prøvast eller falsifiserast, men heller eit rammeverk av teknikkar og metodar for å analysere menneskeleg aktivitet (Kuutti 1996).

”Activity theory lets us study the relationship between the development of the individual and the society in which the person exists”
(Bødker 1996:148)

Historisk sett byrjar AT med klassisk tysk filosofi frå Kant til Hegel i det attande og nittande hundreåret. Aktivitet som konsept vart introdusert i denne perioden. Han vart vidare utvikla av Marx og Engels og seinare av sovjetisk kulturhistorisk-orienterte akademikarar som Leont’ev and Luria (Kuutti 1996). Dei mest grunnleggande prinsippa i AT, i følge Kuutti, er:

- Aktivitetar som grunnleggande analyseining
- Historie og utvikling
- Artefakter og mediering.

Ein aktivitet kan sjåast på som ei dynamisk eining. Aktiviteten er i konstant endring og utvikling. Endringar i aktiviteten skjer ikkje etter faste eller ordna mønster. Det er difor ofte viktig å kjenne bakgrunnen for ein aktivitet, eller sagt på ein anna måte; kjenne dei historiske røtene til aktiviteten. Denne historia må igjen, som alt anna innan AT, tolkast i høve til konteksten.

I AT er analyseininga ein aktivitet rett mot eit mål (objekt) som igjen motiverer handlinga og gir ho ei spesiell retning.

”Activity manifests itself in the form of goal oriented individual *actions* in which the subject is consciously aware of what he or she is trying to accomplish. Action in turn rely on automatic operations, dependent on the conditions at hand.” (Engestrøm 1990:173)

Leont’ev (Centre for CHAT & DWR) identifiserte interaksjon mellom menneske og verda i tre nivå: aktivitetar, handlingar og operasjonar (figur 4), også omtalt som den hierarkiske strukturen til aktivitet:

Level	Oriented towards	Carried out by
ACTIVITY	OBJECT / MOTIVE	COMMUNITY
ACTION	GOAL	INDIVIDUAL OR GROUP
OPERATION	CONDITIONS	ROUTINIZED HUMAN OR MACHINE

Figur 4 Strukturen av aktivitet på tre nivå sett fram av Leont’ev (Centre for CHAT & DWR)

Aktivitetar er styrt av motiv, medvite eller umedvite. Dei manifesterer seg i målretta og medvitne handlingar. Handlingane er igjen bygd opp av automatiske operasjonar. Det skjer stadig overgangar mellom dei ulike nivåa i dette hierarkiet:

”Actions are internalized and become automatic operations through repeated practice. On the other hand, actions may also be expanded into novel collective activities”

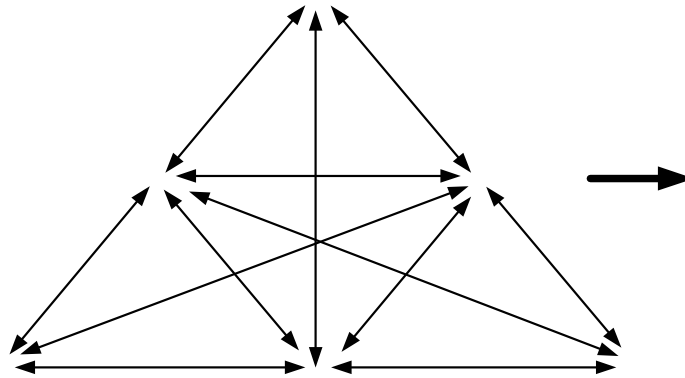
(Engestrøm 1990).

2.5.1 Aktivitetssystemet

Gjennom sitt no vidkjende ”triangel” har Engestrøm systematisert komponentane i aktiviteten i eit aktivitetssystem. Aktivitetssystemet (figur 5) er ein heilskap av seks ulike nodar (Engestrøm 1990).

I aktivitetssystemet finn ein den enkelte aktøren (subjektet), samfunnet, som aktøren er ein del av, og objektet for aktiviteten. Objektet representerer samstundes motivet for aktiviteten. Relasjonen mellom subjektet og samfunnet er mediert av artefakt (verky) i aktiviteten på den eine sida og samfunnet sine reglar for handlingar og

interaksjon på den andre. Samanhengen mellom samfunnet og objektet for aktiviteten er spesifisert gjennom arbeidsdeling. Arbeidsdeling viser kva som vert gjort av kven i samfunnet i høve til det felles objektet.



Figur 5 Engestrøms aktivitetssystem (Engestrøm 1990)

Artefakter har ein sentral posisjon i AT. Eit artefakt er i hovudsak eit verkty. Kor vidt ein definerer det, varierer. Engestrøm presenterer ulike nivå av artefakt og kva rolle dei har. Men det mest sentrale i dette studiet er artefakta si medierende rolle mellom subjekt og objekt. Subjektet arbeider mot objektet ved hjelp av eit artefakt. Desse artefakta er sjeldan av statisk karakter. Artefakt kan seiast og verte skapt og endra i høve til aktiviteten og konteksten dei vert brukt/skapt i. (Engestrøm 1990)

I analysen av aktivitetssystem er det sentralt å ha blick for heilskapen. Dei ulike nodane gjev ikkje meining kvar for seg, men gjennom relasjonane mellom dei. Ein må altså sjå dei i ein samheng. Nodane kan også vera delar av andre aktivitetssystem i ein større heilskap. I AT er ikkje eit aktivitetssystem lukka, men i kontinuerleg "dialog" med andre aktivitetar. Det er og stadige transformasjonar mellom dei interne nodane i eit aktivitetssystem. Konflikatar og motsetnader i og mellom aktivitetssystem er den drivande krafta i denne utviklinga (Bødker 1996.) Ein må ha utviklinga i tankane når ein utførar ein aktivitetsteoretisk studie.

For å skjønna ein aktivitet og ein sosial praksis må vi såleis også kjenna til den historiske utviklinga. AT gjev høve til å ha grupper i eit fellesskap som

Suk

analyseining. AT er ei fleksibel tilnærming som gjev forskaren ein ståstad, eller eit perspektiv i høve til problemområde.

2.6 Læring

”Det som skjer når vi lærer, er at vi skaffer oss evne til å handle med nye intellektuelle og fysiske redskaper som stadig mer kompetente aktører innen for en ny virksomhet.”

(Säljö 2000: 155)

Over vert det presentert ein definisjon av læring, som vil vere relevant i dette studiet av læringseffektar. Gjennom dokumentanalyse og intervju vil eg freiste å avdekke om BU har eksplisitte definisjonar og / eller læringsmål dei ynskjer å oppnå gjennom sitt arbeid. Har dei kompetansemål? Korleis vurderer dei kompetansen til dei tilsette? Vert aukande kompetanse gjennom arbeid vurdert?

Det vil ikkje verte gjort nokon samanliknande studie av læringseffektar, då der ikkje er relevante data å samanlikne med. BU må langt på veg sjølv, gjennom den kvalitative evalueringa som skal gjennomførast, indirekte fortelje kva og kor mykje dei har lært. Særleg gjennom analysearbeidet knytt mot intervju, vil eg kunne avdekke eventuelle endringar i dei tilsette sine læringskurver i forhold til korleis dei har arbeidd tidlegare.

Cockburn (Cockburn 2002) presiserer i si bok at i løpet av eit prosjekt vil utviklarane lære ny teknologi å kjenne, få kjennskap til nye problemområde og prosessar, og lære korleis ein betre skal arbeide saman med nye kollegaer. Ein læringsprosess vil altså alltid finne stad når ein arbeider med eit prosjekt, eg vil freiste å avdekke om det er faktorar ved RUP som gjer læring under denne utviklinga særeigen.

Ut ifrå funna eg har gjort i dette arbeidet, er det naturleg for meg å ta utgangspunkt i ei sosiokulturell forståing av læring. Dei læringseffektane eg meiner å ha funne, oppstår særleg ut ifrå sosialt motiverte faktorar. Eg meiner med dette ikkje å seie at læring berre oppstår i sosiale kontekstar. Den enkelte utviklar vil gjennom heile prosessen, som elles i livet, lære også når han arbeider for seg sjølv.

Lave og Wenger (<http://www.infed.org>) peikar på ein interessant faktor som kan ha verka inn på motivasjon for læring i denne utviklingsprosessen. Dei trekk fram at ein deltakar i eit arbeidssamvirke (community of practice) vert motivert til læring gjennom behov og lyst til å fungere i eit arbeidsfellesskap. Dette meiner eg kan ha spesiell innverknad i høve til innføringa av UML som eit nytt "språk" som skal nyttast for kommunikasjon rundt Forv@lting. Dei tilsette vil verte motiverte til å lære seg dette, for å fungere som fullverdig deltakar i kommunikasjonen knytt mot utviklinga.

I Forv@lting-prosjektet vil dei tilsette vere sett saman i ulike team, som jobbar delvis i fellesskap mot eit felles mål. Det vil her vere interessant korleis dei er i stand til å dra nytte av kvarandre sin kunnskap og kor godt dei klarer å arbeide saman. Det kan her verte aktuelt å dra veksle på Koschman sine teoriar rundt datastøtta samarbeidslæring (Computer Supported Collaborative Learning) (Koschman 1996). Kva verkty, om noko, tek dei i bruk for å sørge for at læring/kompetanse vert tilgjengeleg for dei forskjellige teammedlemmene. Har dei noko verkty som er med å støtter dei i dei ulike prosessane. Aktuelle område kan her vere å spreie nye tankar og idear som kan betre produktet, eller endringar i krav.

2.7 Empiriske systemutviklingsanalysar

Eg vil i dette kapitlet presentere ein del arbeid som er gjort innanfor fagfeltet eg skal ta for meg. Dei fleste av arbeida fell inn under paraplyen kvalitative arbeid knytt mot ein systemutviklingsprosess. Mi eiga metodologiske tilnærming vil verte presentert grundigare i neste kapittel. Eg finn det her fruktbart å presentere korleis andre har gjort det før meg, og korleis dette kan støtte opp om arbeidet mitt og korleis det samtidig har hjelpt meg i utforminga av min forskingsdesign.

2.7.1 Kvalitative studium av systemutvikling

I sin artikkel "Qualitative Methods in Empirical Studies of Software Engineering" (Seaman 1998) gjer Carolyn B Seaman seg til talskvinne for ein meir utstrekt bruk av kvalitativ metode innan datarelatert forskning. Ho viser i denne artikkelen at det er ei stadig aukande interesse og aksept for å arbeide med kvalitative data. Grunnen til dette er at det i den seinare tid har vorte eit stadig skarpere fokus på den menneskelege rolla innan systemutvikling. Når merksemda vender seg frå den tekniske sida av systemutviklinga, og ser meir på menneskelege aspekt, vert ein avhengig av å arbeide ut ifrå andre metodar og forskingsspørsmål. Og i følgje Seaman er det til kvalitativ metode ein skal vende seg i desse tilfella. Seaman hevdar at den aukande aksepten for kvalitativ metode og fokus på menneskelege aspekt kan tolkast i den retning at fagfeltet har nådd eit nytt modningsnivå.

"... software development also presents a number of unique management and organizational issues, or "people problems" that need to be addressed and solved in order for the field to progress" (Seaman 1998: 557)

Fokus på menneskelege aspekt medfører i følgje Seaman at ein i større grad må gå i djupna, og grave i det komplekse problemområdet, snarare enn å freiste å berre trekke ut det som kan verke interessant i utgangspunktet. Dette vert best gjort med kvalitative metodar som i stor grad vart utvikla av forskarar innan pedagogiske og andre sosialt orienterte miljø, for spesifikt å studere kompleksiteten i menneskeleg handling og samhandling. Seaman hevdar vidare at kvalitative data er rikare enn kvantitative. Gjennom å samle inn kvalitative data vil ein dermed sitte på ei auka informasjonsmengd, og eit større mangfald i data.

I artikkelen omtalar Seaman fleire kvalitative metodar for datainnsamling og analyse, og skildrar korleis dei kan nyttast i empiriske studium av systemutvikling. Ho trekker og fram korleis ein i gitte tilfelle med fordel kan nytte kvantitativ og kvalitativ metode i same forskingsprosjekt. I arbeidet mitt vil eg ikkje nytte meg av kvantitativ metode. Forskingsspørsmålet eg ynskjer å finne svar på er av ein slik art at det, slik stoda er no, ikkje vil det vere fruktbart å gjere ei kvantifisering av rådata. I eit seinare eller liknande prosjekt kunne det vere interessant å freiste og forsvare eller bevise funn i dette studiet med kvantitative data frå ei anna eller liknande studie.

Seaman peiker og på negative sider ved å arbeide kvalitativt. Kvalitativ analyse er meir arbeidsintensiv og utslitande enn kvantitativ. Vidare har kvalitative forskingsresultat vorte skulda for å vere for uklare og ”mjuke” i forhold til kvantitative resultat. Kvalitative resultat kan vert oversett då dei ikkje er like enkle og samanfatte og forenkle.

2.7.2 Aktivitetsteori i studium av systemutvikling

I sin artikkel “A view of Software Development Environments based on Activity Theory” (Barthelmess & Anderson 2000) presenterer Barthelmess og Anderson aktivitetsteori som eit godt egna rammeverk for studium av systemutviklingsprosessar.

“Activity Theory is a descriptive tool that is useful for analyzing and understanding collaborative work in general, independently of any specific field of application.”

(Barthelmess & Anderson 2000:2)

Tidlegare var systemutvikling basert på enkeltpersonars arbeid, men etterkvart som teknologi har vorte billigare og meir tilgjengeleg, har det vorte ein aukande etterspurnad etter større og meir kompleks programvare. Systemutvikling har som eit resultat av dette vorte basert på teamarbeid, meir enn på arbeid frå enkeltindivid. Eit problem Barthelmess og Anderson peikar på i høve til dette er at ein ikkje har gode nok teknikkar og verkty for å støtte ein slik kollaborativ prosess.

I artikkelen hevder dei at systemutvikling er ein kollaborativ prosess som tar utgangspunkt i eit systemutviklingsmiljø. Sidan miljøet spelar ei sentral rolle i høve til måten ein samarbeider på i eit systemutviklingsprosjekt, er det sentralt å kunne studere kriteria dei ulike deltakarane samarbeider ut ifrå.

I studia som vert presentert i artikkelen, har Barthelmess og Anderson nytta aktivitetsteori for å analysere og evaluere eit systemutviklingsmiljø kalla Process-centered Software Development Environments (PCSDE's). PCSDE er i korte trekk eit systemutviklingsverktøy som skal støtte samarbeid i store systemutviklingsgrupper.

I sin analyse plasserer dei dei ulike elementa ved PCSDE som fell inn under dei ulike nodane i aktivitetsteorien. (jf 2.5.1) I min analyse vil eg foreta ei liknande dekomponering av RUP slik BU nyttar den. Dermed får eg betre oversikt over korleis dei ulike faktorane i RUP verkar inn på mellom anna samhandlinga mellom dei ulike teama og internt for kvart team.

Susan Bødker (Bødker 1996) har og nytta AT for å studere bruksaktivitet i eit utvida perspektiv. Studiet er konsentrert rundt dei medierande eigenskapane til eit verktøy. Ho argumenterer for aktivitetsteori som eit godt rammeverk for å forstå menneskeleg handling knytt mot informasjonsteknologi. Ho meiner at aktivitetsteori hjelper med å strukturere analysen utan å preskribere kva ein leitar etter. I tillegg vert ein konstant påminna om konteksten og historia til handlingane og operasjonane ein studerer. Dette medfører at ein ikkje vert freista til å forstå situasjonane ein observerer i isolasjon.

I artikkelen nyttar ho omgrepa fokusskifte og breakdowns for å sjå korleis eit artefakt medierer objektet. Eit fokusskifte oppstår i det subjektet ikkje lenger fokuserer på objektet for aktiviteten. Eit slik fokusskifte kan vere frivillig eller ufrivillig. Dersom eit slik fokusskifte er uintendert frå subjektet si side vert det vidare definert som eit breakdown. Tankegangen er at eit verktøy fungerer godt om det lar brukarane rette merksemda mot det verkelege objektet, og dårleg om merksemda vert fjerna frå objektet. Eg har ikkje nytta desse omgrepa i mi analyse,

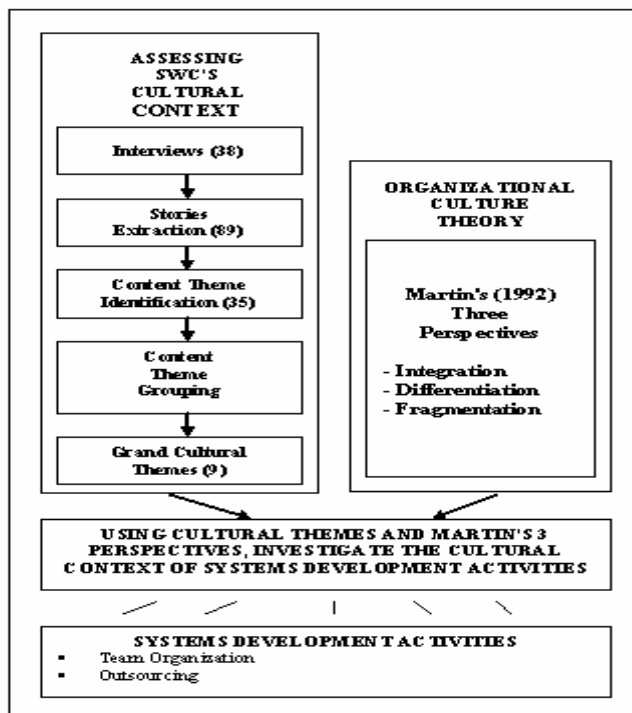
men tankegangen min er ikkje ulik når eg ser etter faktorar som påverkar læring og kommunikasjon ved RUP som metode.

2.7.3 Kulturell kontekst

Dubé og Robey sitt arbeid "Software stories: three cultural perspectives on the organizational practices of software development" (Dubé og Robey 1999) omtalar eit spesifikt case stadium av ein organisasjon som utviklar programvare. Studiet ynskjer å avdekke organisasjonskulturen i ei konkret bedrift som gjennomgjekk store organisatoriske endringar i systemutviklingsprosessen sin. Desse endringane bestod i hovudsak av to nye element; innføring av teamarbeid, og det å sette ut enkelte arbeidsoppgåver til eksterne krefter (outsourcing). Dette er relevant i høve til mitt studium, sidan det er første gongen BU i så stor grad har organisert arbeidsoppgåvene i teamkonstellasjonar som dei gjer i Forv@lting. Utsetjing av arbeidsoppgåver er ei arbeidsform BU er kjend med frå tidlegare arbeid.

Sentralt i studiet til Dubé og Robey er den sosiale og kulturelle konteksten og rolla til den tolkande forskaren. Dette kan i stor grad knytast opp mot det arbeidet eg har gjennomført i høve til BU sitt Forv@lting-prosjekt. I aktivitetsteori, som er rammeverket eg vil nytte i min analyse, står også kulturen eller konteksten dei ulike aktivitetssystema eksisterer innan, svært sentralt.

I figur 6 er forskingsdesignen for arbeidet presentert. Figuren viser gangen i forskinga og kva former for datainnsamling og analyse som har vore gjennomført. Som det går fram av figuren vart det gjennomført 38 intervju, desse var av semistrukturert art. Ut ifrå intervjuet vart det skilt ut 89 "historier." Med historier forstås ein her eit segment frå eit transskript som er av særleg interesse i høve til forskingsspørsmålet. Desse historiene vart så gruppert under ulike tema gjennom ein prosess i tre steg, før dei vart sett opp mot Martins (Martins 1992) tre perspektiv. Gjennom dette siste analysesteget kom dei fram til resultata som vert skissert i nedste del av figuren. I metodekapitlet vil eg presentere ein eigen variant av denne figuren for å synleggjere forskingsdesignen min.



Figur 6 Dubey og Robey si framstilling av eigen metode (Dubé og Robey 1999)

2.7.4 Tolkande casestudium

Walsham ynskjer i sin artikkel "Interpretive case studies in IS research: nature and method" (Walsham 1995) å gi forskarar referansepunkt for korleis ein kan gjennomføre (in depth) casestudium som fokuserer på menneskeleg åtferd særleg knytt mot forskning innan utvikling av informasjonsteknologi.

Artikkelen fokuserer særleg på den vanskelege rolla ein har som forskar i slike studiar. Den viktigaste oppgåva er å gjere sine egne tolkingar av ein annan person si tolking av ein gitt situasjon. Det er i desse situasjonane viktig at forskaren er medviten om rolla si i det som kan beskrivast som ein kompleks menneskeleg prosess. I høve til rapportering av funn i slike studiar er det viktig å presisere at ein ikkje presenterer gitte sanningar. Det som vert presentert, er ein forskar si tolking av andre sine tolkingar. Det er difor viktig i ein rapport å fortelje korleis ein kom fram til dei resultatane ein presenterer.

Walsham hevdar og at det i slik forskning er sentralt å bygge på tidlegare arbeid og teoriar. Dette er med å skape eit teoretisk rammeverk som bygger på etablert kunnskap, og gir ein teoretisk plattform å bygge sitt empiriske arbeid på. Faren med dette kan vere at forskaren ser seg blind på etablerte sanningar, og berre finn det teorien forslår.

Intervju er i følgje Walsham den viktigaste datakjelda i slike casestudium. Gjennom intervjuet får forskaren førstehandstilgang til deltakarane si tolking av faktiske hendingar. Intervjuet er også persistent og kan gjennomanalyserast fleire gonger i ettertid. Det å nytte ei form for opptaksmedium i intervjusituasjonen, kan vere ein fordel, fordi ein får ei fullstendig rapportering av kva som har vorte sagt. Men Walsham presiserer at intervjuet må gjennomførast så "naturleg" som råd. Det er viktig å vere klar over at intervjuobjektet kan verte påverka av opptaksmediet.

Eit sentralt poeng for casestudiet, som Walsham også adresserer, er spørsmålet om i kor stor grad ein kan generalisere ut ifrå eit enkelt casestudium. Generelt kan funn i eit casestudium fungere som forklaring eller støtte for tidlegare studium, mens grunnlaget for å forutsjå eller peike på universelle sanningar er avgrensa. Funna bør oftast sjåast på som ei forklaring av gitte fenomen i ein gitt kontekst, men kan, slik eg ser det, like fullt vere grunnlag eller inspirasjon for vidare forskning der målet kan vere å etterprøve funna i det gitte casestudium.

2.7.5 Metoden si rolle

Det knyter seg skepsis til å bruke ein standardisert metode i eit systemutviklingsprosjekt. Fitzgerald (Fitzgerald 1998) hevdar at ein i litteraturen lenge har hatt eit altfor ukritisk positivt forhold til metodologiar innan systemutvikling, og at der finst fleire argument både for og imot metodebruk.

I artikkelen vert det presentert ei kvantitativ undersøking blant systemutviklarar for å finne kva syn dei hadde på å bruke metodologiar i sitt daglege virke, og kva slags resultat dei kunne vise til. Sjølv om dette arbeidet vart publisert for fem år sidan og er av kvantitativ art, finn eg likefullt interessante poeng å ta med inn i mi eiga forskning.

Fitzgerald hevdar at metodar spelar si mest sentrale rolle i høve til oppdeling, kontroll, struktur og standardisering av prosjekt. Dette kan gi ei prosjektgruppe eit rammeverk og gjerne eit felles sett teknikkar å jobbe ut ifrå. Metodar kan og hjelpe å styre ressursbruken. Det må her nemnast at dette er mykje same argumenttrekke BU nytta for å ta i bruk RUP på Forv@lting-prosjektet. Dei ynskte ein betre struktur og større kontroll over prosjektets utvikling enn dei hadde hatt på tidlegare prosjekt.

Ein interessant påstand frå Fitzgerald er at ingen systemutviklar vil nytte den same metoden identisk. Kvar og ein vil trekke ut det han eller ho meiner er fornuftig og ignorere resten. Vidare vil truleg ingen systemutviklar velje ut dei same elementa frå ein metode, alle vil følgjeleg arbeide litt eller ganske forskjellig. Hans påstand vidare er at dette også vil gjelde for utviklarar som arbeider på det same prosjektet.

Hovudpoenget til Fitzgerald, slik eg ser det, er at ein ikkje må tru at det å ta i bruk ein kjend metode, automatisk vil føre prosjektet trygt i hamn. Alle som ynsker å ta i bruk ein gitt metode i eit prosjekt er avhengig av å tilpasse metoden til prosjektet. På dette punktet får han og støtte av Alistair Cockburn (Cockburn 2002), som kjem med følgjande utsegn i høve til universelle metodar:

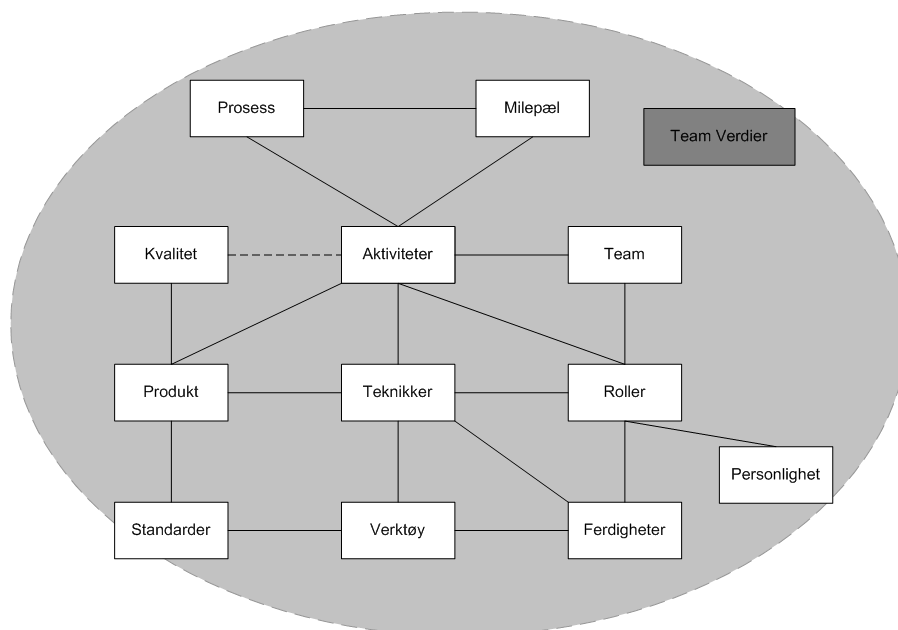
” Don’t even think that Rational Unified Process, Unified Process, or the Something Else methodology will fit your project *out of the box*. If you follow a methodology out of the box, you will have one that fits some project in the world, but probably not yours.”
(Cockburn 2002:172)

Cockburn har, som sitatet over ber preg av, ikkje overdriven tiltru tilstore universelle metodologiar, og representerer for meg ein viktig motpol til Jacobsen, Booch og Rumbaugh. Boka hans har hjulpet meg til å få eit meir nyansert bilete av RUP og UP. Cockburn meiner at ingen systemutviklingsprosjekt er like, og at ein difor alltid må vere opne for forskjellige metodologiar, eller vegar mot målet. Målet bør vere å alltid utvikle eit kvalitetsprodukt på så kort tid som mogleg.

Cockburn hevdar at ein metodologi er alt ein vanlegvis gjer for å ferdigstille ei programvare. Alle organisasjonar har ein metodologi, det er måten dei vanlegvis arbeider på. Metodologien er konvensjonane ei gruppe vert einig om å arbeide etter,

metodologien kan følgeleg sjåast på som ein sosial konstruksjon. Men denne konstruksjonen kan og bør ein revidere og forbetre frå tid til anna.

Cockburn (Cockburn 2002) presenterer ein modell, kalla "elements of a methodology", med 13 element som ein metode skal innehalde (figur 7). Desse elementa koordinerer dei ulike medlemmane av ei prosjektgruppe sin aktivitet. Koordinasjon er eit nøkkelord i Cockburn sine systemutviklingstankar. Ein føresetnad for at gruppa skal lukkast er godt samarbeid og god kommunikasjon i tillegg til koordinasjon; alle sentrale element når ein sosialt konstruerer sin metode.



Figur 7 Cockburn sin "elements of a methodology"(Cockburn 2002).

3 Forskingsspørsmål

Som det går fram av teorien som er presentert så langt i oppgåva, vil eg ha eit fokus på dei menneskelege aspekta ved systemutviklingsprosessen. Sentralt for meg vil det vere at i ein systemutviklingsprosess vil ikkje utviklarane berre utvikle det konkrete prosjekt, men og bygge ny kompetanse. Kva har vorte gjort rett, og kan det byggast vidare på? Kva har ikkje fungert, og kva må betrast til neste gong? Eg legg her til grunn at det alltid vil vere ei form for kompetanseheving i eit systemutviklingsprosjekt, som det jo er i dei fleste situasjonar. For som Säljö (Säljö 2001) hevdar, mennesket kan ikkje unngå å lære.

Dette arbeidet vil vere todelt, først og fremst vil eg sjå utviklingsprosessen som ein læringsprosess både på individnivå og for prosjektgruppa som heile. Sentralt i denne læringsprosessen vil vere kommunikasjonen medlemmene imellom. Eg finn det difor naturleg å inkludere dette som element i forskingsspørsmålet mitt, som har fått følgjande formulering:

Korleis påverkar Rational Unified Process kommunikasjon og læring i systemutviklingsprosessen hjå Bjørgvin Utvikling?

Dette er eit relativt ope forskingsspørsmål, noko som er eit medvite val frå mi side. Eg ynskte å gjennomføra forskinga utifrå ei definert og avgrensa problemstilling, men samstundes innretta slik at eg kunne fange opp andre interessante og relevante funn, sjølv om dei ikkje er direkte knytt til hovudproblemstillinga. Ein slik innfallsvinkel er, slik eg ser det, naturleg for det empiriske casestudiet, som ligg til grunn for denne oppgåva.

I dette studiet vil eg freiste å avdekke i kva grad det i prosjektet skjer ei kompetanseheving for dei tilsette som individ og kollektiv. Eg gjer dette utan ein klar definisjon av kva ei slik kompetanseheving vil innebere. Det vil i datamaterialet vere ei fortolkingsoppgåve (for meg som forskar) å peike på felt der ei slik merkbar kompetanseheving har funne stad. Og like viktig er det å finne ut i kva grad ho er eit

resultat av RUP. Eg vil og freiste å avdekke om RUP på nokon måte har avgrensa kompetansehevinga for prosjektgruppa.

I høve til kommunikasjonsformene som finn stad, vil eg avgrense meg til å sjå på menneskelege relasjonar. Eit spørsmål er i kva grad RUP har vore med å endra eksisterande mønster for kommunikasjon i prosjektgruppa. Vidare må ein spørje om eventuelle endringar har vore forbetringar eller om dei har vore til hinder for prosessen?

Eit anna spørsmål eg søker å finne svar på, er i kva grad utviklingsgruppa nyttar it-verkty for å støtte kommunikasjonen sin. Det vil også vere interessant ut ifrå ein læringsståstad om dei er i stand til å nytte slike verkty for langring og distribuering av ny forståing og informasjon.

4 Forskingsdesign

Dette forskingsprosjektet kan karakteriserast som ein kvalitativ casestudie. Ein casestudie er oftast konsentrert om ein spesifikk førekomst eller situasjon. I dette prosjektet er det Bjørgvin Utvikling sin utviklingsprosess knytt mot utviklinga av Forv@lting som skal analyserast.

Denne studien er i stor grad fokusert på menneskelege faktorar og samspel mellom menneske. Mine funn vil vere av forklarande meir enn av bevisande karakter. Eg ynskjer å beskrive forskingsområdet, og legge fram mine påstandar og teoriar, heller enn å søke etter lovmessigheiter. Difor er det naturleg å gripe an forskinga frå ein kvalitativ ståstad. Seaman hevdar (Seaman 1999) at menneskeleg oppførsel er eit av få fenomen som er så komplekse at det krevst kvalitativ metode for å studere dette.

”It could be argued that human behavior is one of the few phenomena that is complex enough to require qualitative methods to study it.”

(Seaman 1998: 557)

Casestudiet avhjelper kunnskapsbygginga som forskaren søker i forskingsprosessen. Remenyi (Remenyi et al.1998) meiner casestudien oppstår fordi forskaren treng hjelp til å forstå og forklare komplekse fenomen, og at casestudien kan brukast på mange forskjellige måtar og kan tilpassast kompleksiteten. Tankegangen er at ved å fokusere på ein spesifikk førekomst innan forskingsområdet, vil ein kunne presentere teoriar og svar som kan vere nyttige for forskingsområdet som heile. Ved å fokusere på ein spesifikk førekomst vil ein vere i stand til å bringe analysen ned på eit rikt detaljnivå, som kan vere nødvendig for å forstå alle dei samanvevde prosessane som finn stad.

“The case study allows the investigator to concentrate on specific instances in an attempt to identify detailed interactive processes which may be crucial, but which are transparent to the large-scale survey”

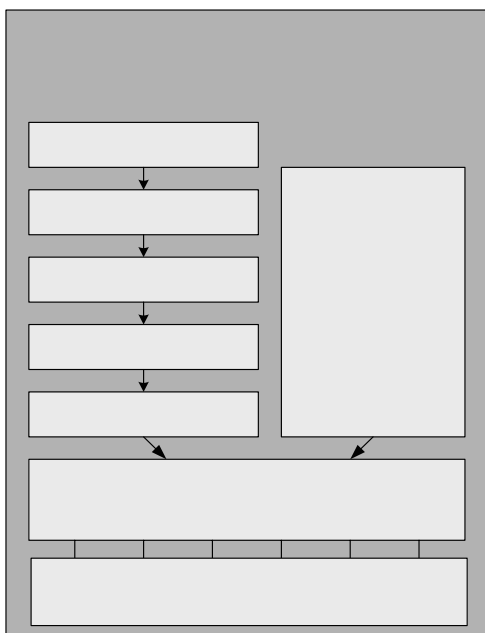
(Remenyi et al.1998:51)

Remenyi (Remenyi et al. 1998) har lista opp fem kriterium han meiner skal finnast i ein eksemplarisk casestudie. Han meiner at ein kan vurdere kvaliteten på studien utifrå kor godt ein følgjer desse prinsippa.

- Han må vere signifikant.
- Han må vere komplett.
- Han bør vurdere alternative perspektiv.
- Han bør legge fram tilstrekkeleg med bevis.
- Han bør framstillast på ein engasjerande måte.

For å illustrere korleis forskningsdesignen og arbeidet med denne oppgåva har skride fram som heile, har eg under plassert mi forskning inn i Dubè og Robey sin figur av korleis eit forskningsdesign kan vere (Figur 8). Bakgrunn og originalfigur vart presentert under punkt 2.7.3 figur 6 (Dubè og Robey 1999).

Figuren syner korleis dei ulike delane i arbeidet bygger på kvarandre, og korleis teorien vert nytta i høve til utkrysstallisering av funn som igjen leier til konklusjonar og resultat for forskinga.



Figur 8 Forskningsdesign fritt etter Dubé og Robey (Dubé og Robey 1999)

4.1 Datainnsamling

Innsamling av data har gått føre seg gjennom ulike fasar og i forskjellige former. Første del av innsamlinga var i form av observasjon og uformelle samtalar. Denne forma for datainnsamling har ikkje vore den mest sentrale av datakjeldene, og vart berre nytta i oppstarten. Den andre kjelda er data som er samla inn gjennom dokumentanalyse, og danna grunnlag for den siste og viktigaste datakjelda, intervju.

4.1.1 Observasjon og uformelle samtalar

Observasjon og uformelle samtalar kan i mange studium vere ei særdeles viktig datakjelde. For ein forskar er det nyttig å vere til stades, og gjennom det få førstehands kjennskap til korleis dei ulike individa arbeider og opptrer individuelt og som eit kollektiv. Forskaren får i tillegg moglegheit til å observere omgivnadene individa arbeider i direkte. Ein kan såleis fange opp moment ved studieobjektet som ein ville hatt vanskar med å fange gjennom andre datainnsamlingsformer, som dokumentanalyse og intervju.

I denne studien har denne forma for datainnsamling i utgangspunktet ikkje stått i høgsetet. Ei moglegheit vi hadde var å gjennomføre delar av skriveprosessen i BU sine lokale. Gjennom dette kunne vi i større grad verte kjent med personane som arbeider der og finne ut korleis miljøet faktisk var. Men dette kunne og ha medført at vi hadde fått eit personleg forhold til bedrifta og tilsette der, noko som igjen kunne ha verka negativt inn på vår objektivitet i høve til studiet. Dette, saman med praktiske implikasjonar, har medført at vi ikkje har nytta deira lokale i skriveprosessen.

I oppstartsfasen av prosjektet hadde eg og medstudent Pilskog fleire møte med kontaktpersonen vår på BU. Desse møta var av ein uformell karakter. Dei gav oss inntrykk av kva type bedrift vi var komen til, samt kva type prosjekt dei var i gang med. Vi fekk og eit innblikk i kva personar som arbeidde der, og korleis tonen og miljøet var m.a. gjennom å delta på ein lunsj med alle tilsette.

I slutfasen av prosjektet har eg saman med medstudent Pilskog presentert forskinga vår både for andre studentar og tilsette internt ved instituttet og for dei tilsette i BU. Særleg presentasjonen hjå BU vil eg trekke fram som ei viktig datakjelde. Skriveprosessen var ikkje fullført, og tilbakemeldingane eg fekk under og etter presentasjonen, har vore med å betre mi forståinga av bedrifta. Desse tente og til å stadfeste og avkrefte tankar eg hadde gjort meg opp i høve til funna i forskinga.

4.1.2 Dokumentanalyse

I denne fasen vart bedrifta sine interne dokumentasjonar av arbeidsprosessar lest og analysert. Denne dokumentasjonen hjelpte meg å danne eit bilete av kva som skjer i bedrifta. Dette var eit tidkrevjande, men viktig arbeid som måtte gjennomførast før arbeidet mot intervjudelen kunne byrje. Remenyi kjem med følgjande påstand i høve til dokumentanalyse:

”They can set the context for interviews or discussions within the organization being studied” (Remenyi et al. 1998:175)

I møte med vår kontaktperson fekk vi lovnad om at vi skulle få rikeleg tilgang til deira interne dokumentasjon av arbeidsprosessar og metodar. Utifrå dette materialet har eg gjennom grundige analysar prøvd å danne gode prosessmodellar og utvide forståinga av bedriftas struktur og arbeidsrutinar. Dette skjedde i samarbeid med medstudent Pilskog.

Det finst ingen enkle metodologiske verkty ein kan nytte på dokumenta ein får tilgang til. Det var ikkje tenleg å nytte noka form for kvantitativ metode som ordteljing av ymse slag. Arbeidet var avhengig av grundig gjennomlesing, sunn fornuft samt innspel frå Pilskog, BU og rettleiar. Mengda av og forma på dokumentasjonen vi fekk overlevert av BU medførte at det ikkje var fornuftig å lese alt frå perm til perm. Arbeidet vart gjort ved å søke etter informasjon som ville vere opplysende i høve til teoriane og dei ulike forskingsspørsmåla. Dokumentasjonen har og i ettertid vorte nytta som oppslagsverk når det har vore nødvendig.

4.1.3 Intervju

Etter å ha opparbeidd ei god forståing av bedrifta sine arbeidsrutinar og metodar, var det naturleg å starte førebuingane til intervju. Intervjua er den viktigaste datakjelda for å svare på forskingsspørsmålet, slik det i følgje Walsham (Walsham 1995) er i dei aller fleste tolkande casestudiar.

Forma på intervju var semistrukturerte uformelle intervju. Ein viktig grunn til å nytte semistrukturerte intervju er at strukturerte intervju fort kan føre til ein unormal og lite konstruktiv dialog. Strukturerte intervju er best eigna når ein arbeider kvantitativt og har ei førestilling om konkrete hypotesar ein ynskjer svar på. Steinar Kvale omtaler semistrukturerte intervju på følgjande måte:

”Det har en rekke temaer som skal dekkes og forslag til spørsmål. Samtidig er det åpent for forandring både i rekkefølge og spørsmålsform, slik at intervjueren kan følge opp svarene og historiene han eller hun får ut av intervjupersonen.”
(Kvale 2001: 72)

Før intervju kunne gjennomførast vart det utarbeidd ein intervjuguide. Ein intervjuguide er meint å indikere område og rekkefølgje for ulike sekvensar ein skal innom i løpet av intervjuet (Kvale 2001). Ein intervjuguide vil følgjeleg innehalde sentrale spørsmål ein ynskjer å få informasjon om, og kva rekkefølgje ein kan tenkje seg å stille dei i. Det er likefullt viktig å vere budd på å avvike frå denne guiden om det skulle dukke opp annan interessant informasjon i løpet av intervjuet. Ein bør og vere budd på å la intervjuobjektet snakke om forskingsområdet slik han ynskjer, sjølv om dette vil avvike frå guiden. Målet er å få informanten på gli, men diskret pense informanten inn på kjerneområda slik at han eller ho med eigne ord skildrar det aktuelle forskingsområdet.

Det var ønskeleg å kunne gjere lydopptak av intervjusituasjonane. Ein stor fordel med lydopptak er at ein slepp å notere under sjølve intervjuet. Dette fjernar eit forstyrrende element i samtalen, og kan gi ein meir avslappa og lett flytande dialog. Eit lydopptak fangar og opp tonen ting vert sagt i, fart, pausar og andre meiningsberande faktorar. Det er og viktig å tenkje over at informanten kan reservere seg i forhold til at alt han eller ho seier vil verte persistent lagra, og kan

ikkje ”rettast opp”. I tillegg kan det hende at informanten ikkje ynskjer at lydopptak vert nytta. Ein må følgjeleg vere førebudd på å kunne klare seg utan lydopptak som innsamlings verkty.

Jordan og Henderson (1995) hevdar at intervjuobjekt ofte raskt vender seg til nærver av videokamera, mikrofonar og liknande. Objektet er ofte bevisst overfor mediet i starten, men ven seg oftast raskt til det og vert ikkje nemneverdig påverka. Denne påstanden var treffande for mitt arbeid. Informantane fokuserte noko på mediet i byrjinga av sesjonane, men etter kort tid verka det som om dei ikkje lenger tenkte over at mediet var tilstades.

Det vart nytta ein minidiskspelar av typen *Sony MZ-G750* med ekstern mikrofon for å ta opp lyd under intervju. Eit minidiskopptak har mange fordelar i forhold til tradisjonelle magnetbandbaserte opptaksmedium som diktafon. Minidisk er eit digitalt format, og kan enkelt overførast til andre lagringsmedium som eksempelvis MP3.

Etter kvar intervju sesjon gjorde eg feltnotat der inntrykka eg sat igjen med etter intervjuet vart notert. Av moment som eg prøvde å få med, var tankar rundt kva eg trudde var sentralt i det objektet hadde å fortelje, og inntrykket av informanten si haldning til det å verte intervju. Desse notata var nyttige å ha i bakhand både når transkriberinga og dataanalysen skulle gjennomførast.

BU sa seg villige til å yte 10 enkelttimar med intervju. Medstudent Pilskog og eg delte desse mellom oss, og skulle i utgangspunktet gjennomføre 5 frittstående intervju kvar.

4.1.4 Transkribering

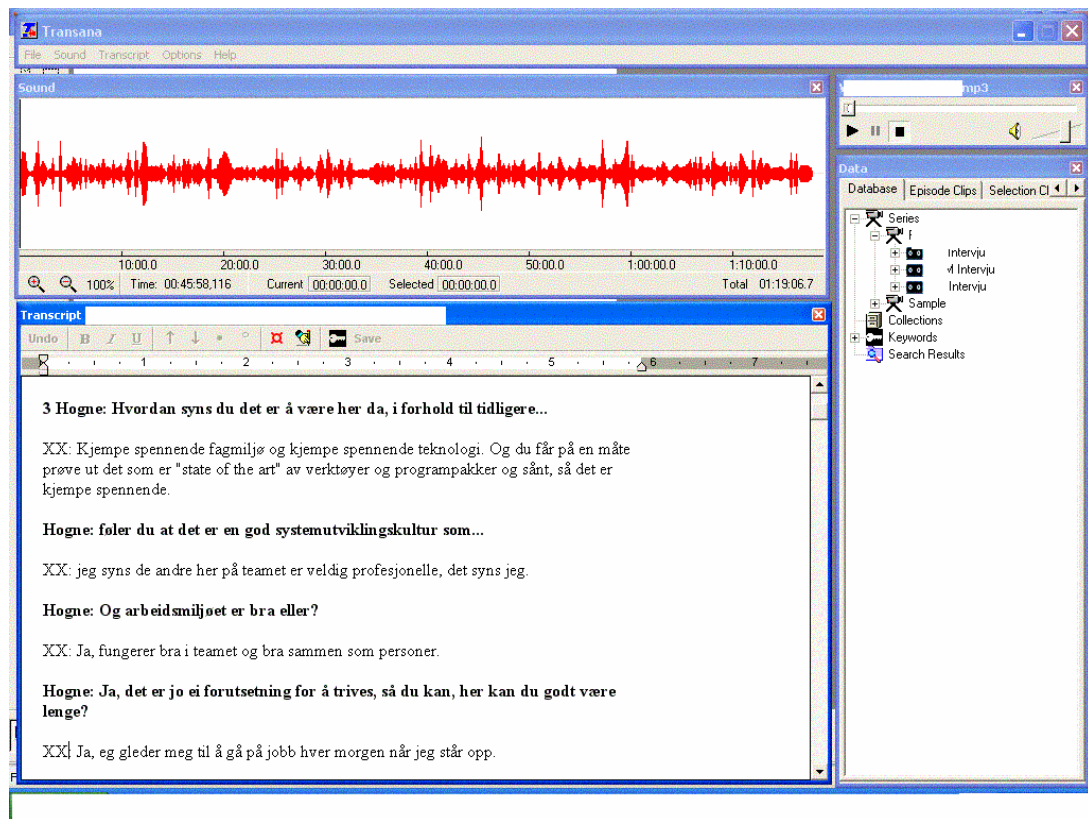
Etter eit intervju var det tid for den tidkrevjande foredlinga av rådata. Intervjua vart overførte frå minidisk format til MP3 før transkriberinga kunne byrje. MP3 vart valt som format av fleire årsaker. MP3 har etterkvart langt på veg vorte eit standardformat for komprimert lydlagring. Faktumet at MP3 er komprimert gjer og at det tek mindre lagringsplass enn om ein skulle lagre det digitalt i reint audio

format, som til dømes Wave. Eit intervju på ca. ein og ein halv time i 128 kbps og 44Khz vil ikkje verte større enn 80-90 MB, som er eit relativt lite og handterlig format. Overføringa frå MD til MP3 vart gjort med eit gratisprogram som heiter SoundCapture (Figur 9). SoundCapture transformerer lydstraumen frå mikrofoninggangen på lydkortet direkte til MP3 format. I figuren under er soundcapture sitt grensesnitt vist, og dei lydinstillingane som vart nytta i overføringsfasen.



Figur 9 SoundCapture sitt grensesnitt med autentiske lydinstillingar

Når intervjuet var i MP3 format vart det importert i Transana (Figur 10). Transana er eit program utvikla for transkribering av lyd og levande bilete, men i dette arbeidet vil berre lyddelen verte nytta. Transana består i hovudsak av fire forskjellige vindauge. Øvst ligg sjølve lydstraumen. Til høgre for den ser ein avspelingsvindauget. Under avspelingsvindauget ligg strukturen til databasen under utarbeiding. Det største feltet er for sjølve transkriberinga. Når ein arbeider i transkriberingsvindauget kan ein styre avspelinga ved hjelp av hurtigtastar. Dette er ein funksjonalitet som lettar transkriberingsarbeidet vesentleg. Delar av skjermbiletet er fjerna av omsyn til teieplikt og for å halde respondenten sin identitet skjult.



Figur 10 Transana sitt grensesnitt med døme på eit transkript og lydstraum.

Der finst fleire retningslinjer for korleis ein skal transkribere eit intervju og kva som skal med. Dette går ikkje minst på kva detaljnivå ein skal leggje seg på, og om heile intervjuet skal transkriberast, eller om ein berre skal trekke ut dei delane ein finn interessante. I dette prosjektet valde vi å transkribere heile intervjujesjonane frå start til slutt. Intervjua vart skrivne på normert bokmål, ikkje på dialekt slik det vart uttalt av intervjuobjektet. Det vart heller ikkje fokusert på non-verbale uttrykk, som tone, fart og pausar, men slike vart like fullt presentere der det var særleg interessant. Det vart i tillegg utarbeidd ein generell mal for korleis transkripta skal sjå ut.

4.1.5 Dataanalyse

”There is little guidance in the literature for the intellectual process of finding patterns and trends in qualitative data.”

(Seaman 1998:567)

Når intervjujesjonane var gjennomført og transkribert var det tid for å ta fatt på arbeidet med analysen.. Dette arbeidet følgde ikkje ein klar og presis metode, sidan

det, som Seaman (Seaman 1998) hevdar over, ikkje finst noka ABC for tolking av kvalitative data. Første del av arbeidet besto i å lese gjennom intervju og trekke ut og tematisere dei sekvensane som verka mest interessante i høve til å svare på forskingsspørsmålet.

Etter at intervjuesjesjonane var gjennomarbeidde, starta arbeidet med å drage ut og formulere funna i forskinga. Ein del av analysen var å plassere BU som bedrift inn i Engestrøms modell for aktivitetssystem, der aktiviteten er å utarbeide Forv@lting. AT fungerte som eit hjelpemiddel for å synleggjere dei funna som vart trekt fram. Alle funn vart og vurdert i høve til andre teoriar samt eigne observasjonar og tankar ein har gjort gjennom forskingsprosessen.

4.2 Forskingsetikk

I eit kvart forskingsprosjekt er der etiske forhold som det må takast omsyn til. I dette arbeidet er det særleg to omsyn som vil vere av relevans. Det første er å handheve teieplikta overfor BU. BU ynskjer ikkje at deira namn eller prosjekt skal kunne verte kjent. Vidare kan det og kome fram informasjon som BU ikkje ynskjer skal verte offentlege i form av "bedriftsløyndomar". Det andre er å respektere intervjuobjektas personvern. Det var eit mål at ingen av objekta i ettertid skal føle at deira respons i resultatform skal verte presentert og teke til inntekt for eit syn dei ikkje kan stå inne for. Elles vil det, som i alle andre vitenskaplege arbeid, vere eit uomtvisteleg ansvar å utarbeide ein rapport der ein ikkje fabrikerer svar det ikkje er grunnlag for, eller på annan måte presenterer urette opplysingar.

5 Evaluering av gjennomføring

I dette kapitlet vil eg ta for meg og evaluere korleis arbeidet med denne forskingsoppgåva har gått føre seg. I førre kapittel presenterte eg korleis eg i samarbeid med rettleiar og medstudent Pilskog planla at arbeidet skulle skride fram. Her vil eg presentere korleis det faktisk gjekk føre seg, og kva erfaringar og vurderingar eg har i forhold til opplegget eg valde.

5.1 Observasjon og uformelle samtalar

Utover dei første møta med kontaktpersonen vår på BU, ein lunsj saman med dei tilsette, og presentasjonen av forskinga hjå BU, har ikkje observasjon og uformelle samtalar vore nytta i særleg grad i dette arbeidet. Eg og medstudent Pilskog hadde tilbod frå BU om å arbeide med oppgåvene våre i deira kontorlokale, i så stor grad vi måtte ønske. Vi har av ulike årsaker valt ikkje å nytte oss av dette tilbodet. Det er reint praktiske grunnar til dette, men vi ønskte også å halde ein kritisk distanse til forskingsobjektet.

5.2 Dokumentanalyse

Vi fekk lovnad tidleg i prosessen om at vi skulle få rikeleg tilgang til BU sin interndokumentasjon. Blant dokumenta som vi fekk tilgang til, var: kravspesifikasjon, løysingsomtale, omgrepsapparat, dokumentasjon av iterasjon 1 og 2, risikoliste, prosjektplan, avviksoversikt, ein CD med ulike figurar og diagram, med meir. Analysen av dette materialet vart gjennomført ved at eg og Pilskog jobba oss gjennom dokumentasjonen frå kvar vår kant. Under denne gjennomlesinga noterte vi oss kva vi fann interessant, og hadde daglege diskusjonar rundt funna vi gjorde og kva inntrykk vi hadde av bedrifta og arbeidsmetodane deira.

Dokumentasjonen var meir omfattande enn det eg i utgangspunktet hadde venta meg. Det var eit tungt arbeid for ein utanforståande å freiste å sette seg inn i BU sine arbeidsrutinar. I dette arbeidet var Engestrøms triangel for aktivitetssystem (figur 5) samt Cockburn sin "elements of a methodology" (figur 7) til hjelp å nytte. Gjennom å fylle ut dei ulike felta i modellane med data frå dokumenta, vart det lettare å få oversikt over BU sine arbeidsmetodar og rutinar.

5.3 Intervjuguide

Då vi vurderte kunnskapane om BU og deira arbeidsmetodar som gode nok, var det tid for å utarbeide intervjuguidar. Under utarbeidinga vart det frå mi side lagt vekt på å få med spørsmål som skulle svare på mitt forskingsspørsmål. Samstundes skulle det gi meg tilstrekkeleg konkret informasjon om BU slik at aktivitetsteori skulle kunne nyttast som rammeverk for mi forståing. Dette arbeidet skjedde også i stor grad i samarbeid med medstudent Pilskog.

Det vart tidleg klart at sjølv om eg og Pilskog skulle ha forskjellig tilnærming i vårt vidare arbeid, hadde intervjuguidane våre mange samanfallande spørsmål. Det vart difor etter konferering med rettleiar bestemt at vi skulle arbeide ut ifrå ein felles intervjuguide og dele datamaterialet. Dette ville auke den enkelte sitt datamateriale, utan å auke arbeidsmengda tilsvarande.

I arbeidet med å samkøyre guidane var det viktig å unngå at intervjuar ville dra for langt ut i tid. Guiden som vart utarbeidd til det første intervjuet, inneheldt heile 32 spørsmål. Denne guiden vart og grunnlaget for dei resterande intervjuar i forskinga. Den generelle guiden er å finne som *vedlegg 1*. Det vart og klart for oss at vi nærma oss meir strukturerte intervju enn det vi i utgangspunktet hadde tenkt. Vi vart og einige om at vi skulle gjennomføre dei første to intervjuar i fellesskap for å sikre at begge sine vinklingar skulle få tilstrekkeleg dekning.

5.4 Gjennomføring av intervju

I utgangspunktet var planen at eg og kollega Pilskog skulle gjennomføre fem separate intervjuer. Som nemnt over, vart vi einige om å arbeide ut ifrå ein felles intervjuguide. I og med at kontaktpersonen vår i første omgang kom opp med åtte intervjuobjekt, vart det totale talet på intervjuer redusert frå ti til åtte. Det vart og bestemt at dei to første intervjuar skulle gjennomførast i fellesskap, og at vi deretter skulle halde tre separate intervju. Under vil det verte presentert korleis eg opplevde dei ulike intervjusituasjonane. Eg legg vekt på respondentens velvilje til å svare på spørsmåla, respondenten sitt syn på forskinga, respondenten si forståing av spørsmåla, intervjuet si form, innhald, verdi, min eigen opptreden og teknisk gjennomføring.

5.4.1 Første intervju

Respondenten var kontaktpersonen vår på BU. Han fungerer som ein *mellomleiar*, med hovudansvar for det tekniske, og er involvert i dei fleste delane av prosjektet.

At det første intervjuet vi skulle gjennomføre var med ein person vi hadde kjennskap til, gjorde det første intervjuet lettare. Respondenten var svært velvillig til å svare på spørsmåla våre, og hadde eit positivt syn på vår rolle som forskarar. Respondenten trakk sjølv fram moment han meinte var sentrale. Intervjuet tok tidleg form av ein dialog der alle dei tre involverte tok ordet der dei fann det naturleg.

Intervjuet strakk seg ut i tid til nærare ein og ein halv time, og respondenten snakka det meste av tida. Vi vurderte etter intervjuet at vi hadde fått gode svar på spørsmåla vi hadde stilt, kanskje for gode. Det var tydeleg at respondenten var begeistra for RUP og at han og var blant dei som bestemte at det var RUP som skulle vere den nye arbeidsmetoden til BU.

Eg var i utgangspunktet relativt godt fornøgd med mi eiga rolle i denne intervjusituasjonen, særleg sidan eg ikkje hadde gjennomført eit vitskapleg intervju tidlegare. Det einaste ankepunktet eg hadde, var at eg nok var litt mykje knytt til intervjuguiden min. Den tekniske delen av opptaket fungerte perfekt og kvaliteten på opptaket var framifrå.

5.4.2 Andre intervju

Vårt andre intervju var med ein *systemutviklar med ansvar for analyse og i særleg grad design*. I tillegg hadde han arbeidd med kravuthenting og ville fokusere meir på etterbehandling av krav i framtidige iterasjonar.

Som den førre respondenten var også denne positivt innstilt til forskinga vår. Han svarte godt og utfyllande på alle spørsmåla og tok sjølv initiativ ved fleire høve. Eg merka tidleg i intervjusesjonen at han vart uroa av at eg tok notat medan Pilskog

intervjua han. Difor fann eg det best å avslutte noteringa og heller gjere fyldige notat etter at intervjujesjonen var avslutta. Dette var og noko eg gjorde i mine resterande intervjusituasjonar. Intervjuet tok etterkvart form av ein relativt naturleg dialog, men glei ikkje like bra som det førre. Like fullt meier eg at vi betre klarte å stille spørsmåla våre der dei passa, og ikkje var like knytt til intervjuguiden som i det første intervjuet.

Vi sa oss begge godt fornøgd etter gjennomføringa av dette intervjuet. Respondenten var reflektert og positiv, og verka noko meir nøytral i forhold til RUP enn førre respondent. Den tekniske kvaliteten på opptaket var også her upåklageleg.

5.4.3 Tredje intervju

Det tredje intervjuet var det første eg gjennomførte aleine. Respondenten var ein person med lang fartstid i BU. Han har i hovudsak *ansvar for programmering på klienten, samt kvalitetssikring av innhald*.

Respondenten var avgrensa villig til å la seg intervju. Då eg ringde han for å avtale ein intervjujesjon, hevda han at han ikkje hadde noko med RUP å gjere og truleg ikkje hadde noko å kome med i høve til forskinga vår.

Det var vanskeleg å få i gang ein like flytande dialog som i dei to føregåande intervjujesjonane. Respondenten svarte som oftast veldig kort på spørsmåla, og tok sjeldan eller aldri sjølv initiativ i løpet av intervjuet. Dette førte til at intervjuet ikkje fekk den flytande forma som dei to første intervjua, og eg vart som følgje av dette veldig knytt mot intervjuguiden, som var kjærkomen og støtte seg til. Utan ein intervjuguide trur eg dette intervjuet hadde vorte lite verdifullt.

Eg hadde inntrykk av at han forsto spørsmåla mine og kvar eg ville, men han meinte sjølv at svara han hadde å kome med ikkje var relevante, då han etter eige utsegn ikkje hadde noko særleg med RUP og gjere. Dette vart det kortaste av intervjua mine, og umiddelbart etter at intervjuet var avslutta følte eg at det hadde vore mislukka. Eg var ikkje nøgd med mine eigne prestasjonar og syntest intervjuet

hadde liten verdi. I etterkant ser eg at respondenten si forståing av ikkje delta i eit RUP-prosjekt er av like stor analytisk verdi for meg, som om han skulle ha vore ein ivrig tilhengjar.

Den tekniske gjennomføringa av dette intervjuet var heller ikkje særleg vellukka. Når eg etter intervjuet skulle høyre på resultatet, var der ei konstant forstyrning på lydstraumen. Eg er ikkje sikker på kva som var grunnen til denne forstyrninga, men ein teori er at datamaskina som stod på bordet der mikrofonen låg, kan ha verka inn. Ein annan faktor som gjorde kvaliteten på opptaket endå dårlegare, var at respondenten ikkje satt stille og ofte lente seg vekk frå mikrofonen, slik at det vart vanskeleg å høyre kva han sa.

5.4.4 Fjerde intervju

Mitt andre solointervju vart gjennomført med relativt nyttilsett person. *Han arbeidde i hovudsak på programmering av tenaren, men skulle i neste iterasjon få nokre oppgåver knytt til analyse/design*

Respondenten var svært villig til å svare på spørsmåla eg stilte. Det verka og som han såg ein verdi i det forskingsarbeidet eg dreiv med. Respondenten svara utfyllande på alle spørsmål og tok sjølv opp tråden fleire gonger og poengterte det han syntest var sentralt og viktig.

Under denne intervjujesjonen opplevde eg at eg ikkje lenger vart for knytt til intervjuguiden. Eg var flinkare til å stille spørsmåla mine der dei passa inn i samtalen, ikkje nødvendigvis etter rekkefølga eg hadde skrive dei i på førehand.

Den tekniske gjennomføringa var det ingen ting å seie på ved dette intervjuet. Klok av skade testa eg kvaliteten eg ville få på opptaket ved å be respondenten snakke i sitt naturlege stemmeleie medan eg tok eit testopptak som eg kontrollerte var av god kvalitet før sjølv intervjuet starta.

5.4.5 Femte intervju

Dette intervjuet vart aldri gjennomført. Etter fleire utsetjingar og forsøk på å nå respondenten m.a. pr e-post fekk eg til slutt beskjed frå kontaktpersonen vår på BU at denne respondenten var særdeles opptatt for tida, og hadde problem med å ta seg tid til ein intervjujesjon. Kontaktpersonen vår spurde vidare om vi hadde gjennomført tilstrekkeleg med intervju, eller om han måtte finne ein ny respondent.

Det hadde no gått 2-3 veker sidan medstudent Pilskog hadde gjennomført sitt siste og femte intervju. Etter å ha konferert med han og rettleiar, kom vi fram til at dei til saman sju intervjujesjonane vi no hadde, var tilstrekkeleg til å starte på sjølve analysefasen.

Eg vil her ikkje spekulere for mykje i kva som kan vere årsak til at denne respondenten ikkje ville la seg intervju. Ein årsak kan vere at denne respondenten ikkje fann det vi dreiv med som viktig nok til å sette av tid til det. Det må her skytast inn at dette var ein person som ikkje var direkte tilsett i BU, men ein av konsulentane. Sjølv om det er uheldig for mengda av analysedata, er det like fullt ei erfaring og ta med seg vidare. Ikkje alt går slik ein planlegg det, og ein kan møte motstand eller tilbakeslag på dei fleste område. Det totale talet på intervju vart no redusert til sju.

5.4.6 Pilskog sine intervju.

I tillegg til dei overnemnde intervju har medstudent Pilskog gjennomført tre intervju åleine. Eg vil ikkje her i særleg grad gå inn på korleis han har opplevd desse intervju eller kva hovudinntrykk han sitt at med. For utfylljande informasjon kring desse intervju viser eg til Pilskog sin rapport (Pilskog 2003) Personane han intervju var:

- Konsulent med ansvar for analyse og design. Hovudfelt design
- Systemutviklar med hovudvekt på teknologisida/programmering. Arbeider også innan databaseadministrasjon og ein del drift og oppfølging av system.

- Innleigd konsulent med hovudansvar for programmering, i tillegg til andre oppgåver som design, arkitektur, konfigurasjonsstyring og test.

5.5 Formatering og transkribering av intervju

Etter at det første intervjuet var gjennomført, var det tid for å overføre lydopptaka frå Mini Disk til MP3 format som er lettare å arbeide med på ei datamaskin. Det vart eksperimentert med korleis ein best skulle lagre og organisere lydfilene. I første forsøk vart kvart spørsmål lagra som ei eiga mp3 fil, og alle filene lagra i egne mapper. Dette var tungvint, og intervjuet vart meir komplekse å handtere i Transana. Vi fann at det hadde størst nytteverdi å lagre intervjuet i ein heil lydstraum.

Sjølve transkriberinga var like tidkrevjande som vi hadde frykta. Sidan det skulle vere eit felles datamateriale, måtte det avtalast ein felles mal for transkriberinga. Vi vart tidleg einige om å transkribere intervjuet i full lengde. Dette medførte at ikkje berre dei delane som på dette tidspunkt verka sentrale skulle transkriberast. Ein grunn til dette valet var at dei to oppgåvene har forskjellige vitskapelege fokus, og at det ikkje er lett å vite så tidleg i arbeidet kva som kan verte viktig seinare. Transkripta skulle vidare vere på bokmål og utan dialektord, men vi ville ikkje i særleg grad korrigere oppbygning av setningar, då sjølve framføringa kan vere meiningsberande i seg sjølv. Transkripta er såleis noko avvikande frå intervjuet, men vi har vore særskilt opptekne av at det ikkje skulle gå på kostnad av innhald.

5.6 Første gjennomlesing, analysestart

Då datamaterialet var ferdig utarbeidd og klart for analysering, byrja dei første gjennomlesingane av transkripta. For min eigen del byrja eg då øvst i bunken med ein markør i handa og las meg gjennom heile bunken fleire gonger. Heile tida las eg med forskningsspørsmålet og dei tankane eg hadde gjort meg, i bakhovudet. Samstundes var det sentralt, å dra ut tilstrekkeleg med data frå materialet til å kunne starte på den verkelege analysefasen der aktivitetsteorien står sentralt.

6 Analyse

Som eit ledd i analysearbeidet vil eg her plassere BU som bedrift inn i Engestrøms modell for eit aktivitetssystem (figur 5). Dei data eg sit på etter datainnsamlinga avgjer kva node elementa frå utviklingsprosessen er plassert på. Når utviklingsprosessen som heile er plassert inn i eit aktivitetssystem, vil det gi eit innblikk i korleis BU arbeider. Abstraksjonsnivået vil i utgangspunktet vere ganske høgt, men eg vil og under kvart punkt gå meir i djupna for å trekke fram og diskutere element som er av særleg interesse, og som kan vere med på betre forståinga av BU sin systemutviklingsaktivitet.

6.1 *Utviklingsprosessen som eit aktivitetssystem*

6.1.1 **Subjekt**

Kva som skal definerast som subjekt i eit aktivitetssystem er avhengig av kva abstraksjonsnivå ein legg seg på. Sidan eg her har valt å modellere prosjektutviklinga som heile, vil det her vere *prosjektgruppa* som heile, som utgjer subjektet. I sitt arbeid med Forv@lting har BU delt opp arbeidsstokken i ulike team, der kvart team har relativt spesifiserte arbeidsoppgåver. Gitt eit lågare abstraksjonsnivå der ein eksempelvis fokuserer på design som aktivitet, ville subjektet vorte designteamet der deltakarane i ei viss grad har overlappende roller. Det er og eit alternativ å definere kvart einskild medlem i gruppa som eit subjekt, men i høve til analysenivået eg har valt, finn eg det mest passande å definere gruppa som heile som subjektet.

Det er og ei moglegheit å knytte fleire personar og element til subjektnoden. I BU som bedrift er det fleire tilsette som ikkje er medlemmer av prosjektgruppa, men likevel har interesse av eller vert påverka av systemutviklingsprosessen. Desse deler og prosjektgruppas mål i form av at prosjektet skal verte ferdigstilt i tide og vere av god kvalitet. Men eg finn det meir passande å plassere dei inn under samfunnsnoden.

6.1.2 Objekt

Det overordna objektet når ein ser på systemutviklinga generelt, er Forv@lting. Det kollektive målet for systemutviklingsgruppa er utvilsamt ferdigstilling av prosjektet i løpet av 2005. Når dei når dette målet, har dei et resultat, eit handfast objekt dei kan overlevere oppdragsgivar.

Som under subjektet er objektets nemning avhengig av abstraksjonsnivået. I eit kvart aktivitetssystem kan objektet brytast ned i fleire delobjekt. Objektet er heller ikkje nødvendigvis statisk, men kan vere i endring gjennom heile aktiviteten. Sidan eg i denne forskinga ser utviklingsprosessen som ein læringsprosess, vil også læring vere eit objekt i aktiviteten. Eit anna objekt i systemutviklingsprosessen til BU er innføringa av den nye metoden, ein modifisert RUP, som dei nyttar for første gang. Fleire av intervjuobjekta, særleg dei som hadde arbeidsoppgåver knytt mot analyse og designarbeidet, peika på kor viktig det var å få metoden til å sitte. Å bruke tid på å tilpasse den til organisasjonen var eit mål i seg sjølv, særleg i tidlege iterasjonar av utviklingsprosessen.

Om ein bryt ned objektet til individnivå, kjem det i fleire av intervjua fram at personar hadde egne delmål som ikkje var direkte knytt mot prosjektet i seg sjølv. Ein respondent uttrykt mellom anna at han hadde eit mål om å lære seg Enterprise Java. Men i dette arbeidet vil det i liten grad verte fokusert på desse.

6.1.3 Verkty

I dette studiet vil eg særleg fokusere på to nivå eller typar av verkty, handfaste og abstrakte. Dei handfaste verktya er dei som er lettast å få auge på. Under er dei mest sentrale handfaste verktya presentert punktvis med tilhøyrande bruksområde:

- MS Office pakke → Kontorstøtte og utarbeidinga av ulike dokument
- MS Visio → Utarbeiding av modellar, generelt teikneverkty
- Magic Draw → Modellering
- Delphi → Prototyping
- Power Designer → Data modellering

- Quick View → Time registrering
- JDK frå Sun → Programmering og kode
- Idea frå IntelliJ → Programmering og kode
- Junit →Automatiske testar
- Bugzilla → Feilhandtering
- Intern Web → Lite brukt så langt
- RUP

I organisasjonen som heile vert det sikkert nytta fleire ulike andre handfaste verkty i utviklingsprosessen. Men for subjektet, slik det er definert over, er dette dei mest sentrale handfaste verktya i følgje intervjua og dokumentanalysen som har vorte gjennomført.

Der er ulike årsaker til at dei overnemnde verktya er vorte valde, men ein faktor som fleire av intervjuobjekta peikar på, er økonomi. BU er ikkje ei bedrift som slit økonomisk, men dei har ein kultur som tilseier at ein ikkje skal sløse. Dei har difor valt billeg eller gratis programvare, framfor store kjende og dyre løysingar der det har latt seg gjere. Det må nemnast at dei har kjøpt nokre lisensar på Rational Rose som er eit komplett systemutviklingsverkty, som skal støtte heile systemutviklingsprosessen. Men BU har vurdert det slik at det er for dyrt i høve til nytteverdien det gjev, om ein skulle kjøpe lisensar til heile arbeidsstokken.

Av dei mest synlege investeringane på verktysida er innkjøp av sett malar frå Rational Software Company, som i veldig stor grad ligg til grunn for systemutviklinga. Men å plassere dette settet med malar som eit verkty, er ikkje heilt ukomplisert. For ein kan og hevde at dette settet med malar er eit sett reglar subjektet vert tvinga til å arbeide etter og ikkje eit verkty dei nyttar. Dette vil verte handsama grundigare i kapittel 7.

Kjennskap til verkty eller eit verkty si historie i høve til subjektet, er og eit moment som må takast med. Fleire av verktya som er lista opp ovanfor, er verkty som dei har nytta i tidlegare prosjekt eller av annan grunn har kjennskap til. Verkty har, som omtalt i teorikapitlet, ei medierande rolle mellom subjektet og objektet. Korleis

verktøyet medierer vil vere avhengig av kva forhold subjektet har til verktøyet. Forholdet subjektet har til verktøyet vil oftast vere i endring over tid, og eit heilt nytt verkty kan endre denne medieringa i høve til objektet på godt og vondt.

I dette prosjektet vil eg hevde at forholda mellom subjekt og verkty av handfast karakter så langt har vore relativt statisk. Det har ikkje vorte innført nye verkty så langt i prosessen som har hatt stor innverknad, og verktya som vart valt i byrjinga, har fungert på ein tilfredsstillande og kjend måte.

Like sentralt, om ikkje meir, står dei meir abstrakte verktya subjektet nyttar i systemutviklinga. Av særleg interesse er kommunikasjonsformer og andre interaksjonsformer og mønster. Det meste av kommunikasjonen i aktiviteten går føre seg i innarbeidde uformelle munnlege former. I dette prosjektet har ein prøvd å formalisere kommunikasjonen og gjere han meir persistent. Dette har ein prøvd å få til gjennom å publisere idear og endringsforslag elektronisk på ein intern arbeidsweb. Dette har berre lukkast delvis. Ingen har uttrykt misnøye med forma den uformelle kommunikasjonen har. Det vert snarare sett på som ein styrke at dei kan kommunisere fritt og ope om det meste, men enkelte ville kanskje sett at den i enkelte samanhengar burde formaliserast noko, slik at vedtekne endringar ikkje vert gløymt. Dei har handfast verktøystøtte for slik endringshandsaming, men har ikkje teke den godt nok i bruk.

6.1.4 Samfunn

Som det vart nemnt under diskusjonen i høve til kven som kjem inn under subjektnoden, er det fleire personar og organisasjonar som har eigeninteresser i høve til objektet. Dei som fall utanfor subjektnoden er det naturlege å plassere her. Det gjeld til dømes tilsette i BU som ikkje direkte sit i prosjektgruppa, oppdragsgivar og staten, representert ved eit departement. I tillegg til departementet, vil også Stortinget og politikarane verte ein del av samfunnet. Lovendringar og andre vedtak som vert fatta i Stortinget i tilknytning til fagfeltet Forv@lting skal handsame, kan i mange høve få direkte følgjer for systemet under utvikling.

I tillegg til departementet, som er oppdragsgivar, er og dei ulike sluttbrukargruppene del av samfunnet på dette abstraksjonsnivået. Sluttbrukarane er involverte både i kravfasen og i kvar av dei ulike testfasane og kjem gjennom heile prosessen med ynskjer og krav som BU må relatere seg til. Dette påverkar heilt klart systemutviklings-aktiviteten.

Ein annan del av samfunnet som og bør takast med, er dei ulike firma og konsulentbyrå som BU direkte har relasjon til. Eg tenker særleg på dei firma som har konsulentar direkte tilsett på Forv@lting-prosjektet.

Med utgangspunkt i prosjektets storleik og tidsaspekt vil det og kunne hevdast at bransjen som heile kan vere ein del av samfunnet. IKT bransjen som heile er av ein relativt dynamisk natur og stadig i utvikling. Det er ikkje usannsynleg at teknologiske og teoretiske nytenkingar i bransjen kan ha innverknad på systemutviklingsaktiviteten

6.1.5 Reglar

Under analysen av arbeidet med Forv@ltingprosjektet har det ikkje kome fram særleg mange formelt definerte reglar for korleis arbeidet på BU skal organiserast. Det meste av reglane som ligg til grunn for den daglege drifta, er innarbeidde normer og konvensjonar som ein finn på ein kvar arbeidsplass. Ein møter presis, gjer arbeidet sitt og følgjer vanlege sosiale normer for menneskeleg omgang.

Av spesielle konvensjonar og normer for BU må nemnast den uformelle og munnlege kommunikasjonsforma, som er nemnt over. Men eit anna interessant aspekt ved BU er at dei har ein relativt flat organisasjonsstruktur. Ein slikt flatstruktur gjer det mogleg for alle i systemutviklingsgruppa å verte høyrte.

Organisasjonsstrukturen gir og moglegheiter for alle til å kome med sine innspel og verte høyrte. Der er ei formelt definert leiing for prosjektet, men det verker som om gruppa som heile heller vil løyse usemjer og fordeling av oppgåver seg i mellom, enn at ein person skal sitte på toppen og bestemme. Om nokon har endringsforslag, treng han ikkje å gå "tenesteveg" for å ta det opp, men kan dele sine tankar med dei

andre på det faste måndagsmøtet. Måndagsmøtet er eit av dei faste haldepunkta for systemutviklingsgruppa i dette prosjektet. Kvar måndag morgon skal dei ulike teama presentere kva dei har gjort og kva dei planlegg å gjere framover. Dette er og forum for endringsforslag og andre innspel i høve til systemutviklinga.

Dei tilsette som vi har snakka med er delt i synet på møtefrekvensen i dette prosjektet. I tillegg til dei faste måndagsmøta held og dei ulike teama eigne møte i forhold til sine oppgåver. Nokre meiner at møtefrekvensen er viktig i høve til å sikre kvaliteten på produktet, i så måte særleg måndagsmøta. Andre igjen finn at frekvensen av møte er for høg. Dei ser ikkje verdien i all pratinga, men ynskjer heller å faktisk gjere noko.

6.1.6 Arbeidsdeling

Ordet team har vorte nemnt nokre gonger allereie. I BU har dei delt arbeidsstokken i ulike team med ansvar for forskjellige delar av systemutviklinga. Teama varierer i storleik, men i følge dokumentasjonen vi fekk overlevert knytt mot dei første iterasjonane, skulle storleiken vere frå 3-6 personar.

Sjølve inndelinga av teama, dvs kven som skal gjere kva, skjedde mykje på grunnlag av kva den enkelte sjølv ønskte å gjere. Dette må sjåast i forhold til den flate strukturen bedrifta har. Alle kan sjølvsagt ikkje få viljen sin heile tida, men dei strevar etter at dei fleste skal få arbeide med det dei ynskjer. Men går ikkje kabalen opp, vil leiinga skjere igjennom.

Ein annan faktor ved teama er at fleire personar kan kle ulike roller og vere involvert i fleire team på same tid. Det er og eksempel på at personar skifter heilt eller delvis roller i løpet av utviklingsprosessen.

6.1.7 Aktivitetsnivå

Som omtalt i teoridelen kan ein analytisk dele subjektets veg mot objektet inn i tre nivå; aktivitet, handling og operasjon. Desse nivåa refererer til graden av medvit subjektet har i høve til aktiviteten.

I aktivitetssystemet som er omtalt over, vil det vere rett å kalle dei ulike arbeidsflytane for aktivitetar. Dei ulike arbeidsflytane som er presentert er: krav, analyse, design, implementering og testing. Kvar av desse aktivitetane er knytt opp til ulike team som har som mål for å ferdigstille sine oppgåver gjennom kvar iterasjon.

Handlingar er meir spesifikke delar av det som totalt utgjer ein aktivitet. I kvar iterasjon er det konkrete ting, ofte gjentakande for kvar iterasjon den enkelte person og team må utføre for å fullføre den gitte aktivitet i kvar iterasjon. Eksempel på slike handlingar kan vere: Planlegging, design av use case, realisering av use case, implementering av use case, utvikle systemtest, køyre systemtest etc.

Det siste og lågaste nivå vert omtalt som operasjonar. Med operasjonar vil ein her forstå det ein konkret utfører for å realisere ei handling. Dette kan vere ting som tastetrykk, museklikk, skrive ut, lagre, etc. I dette studiet vil det analytiske fokus ikkje ligge på dette nivået, då det ikkje vil vere til særleg hjelp for å svare på forskingsspørsmålet.

7 Funn og observasjonar

I dette kapitlet vil funn og generelle observasjonar eg har gjort gjennom dette forskingsarbeidet, verte presentert. I hovudsak vil eg her svare på forskingsspørsmålet som vart stilt ved inngangen til forskinga. Eg vil og gje rom for å presentere fleire interessante aspekt ved metoden og bedrifta som har vorte synlege gjennom arbeidet. Funna er delt opp under delvis overlappande tema, med spesifikke hovudpunkt.

Eg vil først ta for meg BU generelt og beskrive dei tankar og inntrykk eg sit igjen med etter arbeidet mitt. Dette dannar grunnlaget for mi forståing av firmaet og måten dei arbeider på generelt. Under dette punktet vil eg legge fram korleis BU har innført og teke i bruk RUP som systemutviklingsmetode. Det vil her og verte presentert korleis dei tilsette har opplevd å arbeide med iterasjonar.

I neste hovudpunkt vil eg ta for meg korleis dei tilsette har endra si oppfatning av RUP i løpet av systemutviklingsprosessen. Dokumentasjon og særleg kommunikasjon vil vere det neste emnet. Til slutt vil eg ta for meg læringseffektar. Dette emnet har eg vidare delt inn i tre kategoriar: domenekunnskap, metodekunnskap og teknologisk kunnskap.

Det er her på sin plass å nemne at arbeidet er av ein kvalitativ utforskande art. Påstandane som kjem fram under, bør ikkje lesast som fullgode sanningar. Dei er mine tolkingar av respondentane si oppfatning av sin eigen arbeidssituasjon. Om ein ynskjer å finne sanningar eller etterprøve påstandane eg kjem med, må det gjerast ein vidare, gjerne samanliknande, studie av same art og større omfang. Det vil i presentasjonen verte nytta aktivitetsteoretiske termar, som omtalt tidlegare i rapporten.

7.1 BU og UP generelt

BU er ei mellomstor bedrift med 35 tilsette. Organisasjonen er delt mellom eit kontor i Bergen og eit i Oslo. Fleire av dei tilsette i Bergen var tidlegare tilsett i eit konsulentfirma som i stor grad arbeidde for BU. Bedrifta har ein flat struktur der dei fleste kan fremje sine meiningar. Alle våre respondentar gav uttrykk for at dei meinte miljøet var triveleg, humørfyllt og prega av uformell kommunikasjon. Dette er og eit inntrykk eg og medstudent Pilskog sit igjen med etter den kontakta vi har hatt med BU. Ut ifrå mine opplevingar og intervju er det naturleg å hevde at BU har det ein må kunne kalle eit godt arbeidsmiljø, der dei tilsette kjenner kvarandre sine sider på godt og vondt.

”Så er vi en fin sammensetning og fyller hverandre veldig godt... Jeg tror vi er en veldig fin gjeng!”

Mellomleder, med hovudansvar for det tekniske.

”Vi har vel fått en sånn rangordning og hakker på hverandre på godt og vondt, så godt kjenner vi hverandre. Vi gjør det, og det er kanskje en del av svakheten også, men samtidig er det styrken vår når det fungerer”

Mellomleder, med hovudansvar for det tekniske.

Ein fordel med ein slik tett og samansveisa organisasjon er at ein har god oversikt over dei einskilde sine kjerneområde for kompetanse. I dei fleste intervju kjem det klårt fram at når dei møter konkrete problem, til dømes juridiske spørsmål, veit dei kven dei skal vende seg til for å finne svara dei søker. At ein kjenner sine egne og andre sine kompetanseområde godt, lettar og arbeidet med å fordele personar i dei ulike teama med forskjellig ansvarsområde.

Eit særtrekk ved BU er at dei baserer seg på å leige inn kompetanse på dei områda der dei ikkje har god nok dekning. I dette prosjektet har det vorte leigd inn to konsulentar. Dei har særleg hatt i oppgåve å yte teknisk kompetanse for å realisere prosjektet. Ei slik outsourcing av oppgåver kan vere problematisk, men det verkar ikkje å vere tilfelle for BU. Det verkar som om dei innleidde kjem godt overeins med og kommuniserer godt med dei vanleg tilsette. For ei noko grundigare drøfting

av effektane av outsourcing av arbeidsoppgåver, sjå Espen Pilskog sitt arbeid (Pilskog 2003)

Det er verdt å nemne at fleire av dei tilsette var uavhengige konsulentar i eit eige selskap som i stor utstrekning jobba for BU, før dei vart tekne opp i BU for to år sidan. Dei er med andre ord ikkje ukjende med ei slik arbeidsform. Kulturen for å leige inn kompetanse var og ein av grunnane til at RUP vart vald som systemutviklingsmetode. RUP vart vald fordi BU då visste at dei kunne leige inn kompetanse om dei møtte problem knytt til metode og prosess dei sjølve ikkje klarte å handtere.

”Det var vel svært få av oss som begynte på prosjektet som hadde noe særlig kjennskap til det (RUP) på forhånd, og det er jo en stor risiko i seg selv, og da var vi avhengige av at vi kunne kjøpe konsulenter hvis vi hadde behov for det”

Mellomleder, med hovudansvar for det tekniske.

7.1.1 Tilpasse metoden til prosjektet

RUP vert kritisert av mellom andre av Cockburn (Cockburn 2002) for å vere for omstendig og fokusert på modellering, i høve til vinsten med all modellering og krav til dokumentasjon. Han hevdar og som omtalt under punkt 2.7.5, at ein universell metode aldri vil passe alle prosjekt. For BU sin del føler eg at denne kritikken mot RUP, slik dei nyttar den, ikkje er særleg treffande. Gjennom analysearbeidet har eg funne fleire prov på at BU ser ut til å ha vore medvitne i høve til ikkje å la metoden styre prosjektet, men trekke ut viktige delar og tilpasse metoden. Det har sidan starten vore snakk om at metoden deira er ein modifisert RUP. BU har kjøpt eit sett malar frå Rational og plukka ut det dei meinte var fornuftig for deira bedrift i dette prosjektet. Dette er ei metodetilpassing som Cockburn er einig i. Han hevdar at dette er enklare enn å skulle lage ein ny metode frå byrjinga og meir effektivt enn å nytte ein metode som er tilpassa eit anna formål. (Cockburn 2002)

”... vi har ikke valgt å følge slavisk Rational sitt opplegg med støtteverktøyer og alt mulig. Vi har kun valgt å bruke ett malsett, og vi ser på hvordan det er strukturert og så tilpasser vi det til det vi mener er vårt behov.”

Mellomleder, med hovud ansvar for det tekniske

Om BU skulle brukt RUP heilt etter boka, slik det vert lagt opp til frå Rational Software Company, ville det slik eg og fleire av informantane ser det, vore ein stor fare for at BU hadde dokumentert seg sjølve vekk. Fleire av dei tilsett uttrykte at dei på førehand frykta at RUP skulle verte ei stor uhandterleg papirmølle som var meir til hinder enn til nytte, og at det ville vere til hinder med tanke på kreativitet.

”Det var en del skepsis at vi her pådro oss en stor papirmølle som vi ikke kom oss ut av osv”
Systemutviklar med ansvar for analyse og i særleg grad design.

”Jeg har ikke følt at metoden legger noe hinder på min kreativitet, at nå må jeg stoppe å programmere og skriv masse papir”
Programmerar i hovudsak på tenar.

Det går klart fram av BU sin interne dokumentasjon at dei har eit bevisst forhold til å nytte RUP på sin eigen måte. Å seie heilt konkret kva element som er med og kva som er utelate frå RUP i BU sin modifiserte versjon, kjem ikkje krystallklårt fram ut ifrå mitt datamateriale. Men eit hovudmål for BU har vore å nytte prosesstankegangen som ligg bak RUP på ein fornuftig måte. I all hovudsak vil dei følgje hovudelementa i RUP som omtalt i teoridelen. Punktvis kan behovet og motiva for å nytte ein på førehand skildra prosess presenterast slik:

- Dei skal sjølve definere korleis deira eigen prosess skal vere.
- Dei har mange uerfarne personar knytt til objektorientering.
- Utviklingsprosjektet skal relatere seg til mange ukjende teknologiar.
- Realiseringa skal gjerast på ein ny og ukjend måte.
- Det er eit ønske om å skape tryggleik og felles forståing i prosjektgruppa for korleis prosjektet skal gjennomførast.
- Dei har eit ønske om å minske risiko.

7.1.2 Iterasjonar

Eit av særtrekka ved RUP er at prosjektet som heile vert delt i fleire miniprojekt kalla iterasjonar. Å dele opp eit prosjekt og arbeide mot gjevne mål på gjevne datoar er ikkje noko nytt. BU har tidlegare i større prosjekt arbeidd med klassiske milepælar. Iterasjonane fungerer og som milepælar, men er og noko meir. Hovudskilnaden er at ved kvar iterasjonsslutt skal det vere utarbeidd eit inkrement, ein fungerande og køyrbar tilvekst til systemet under utvikling.

”Det er noe annet, fordi når en iterasjon er ferdig så sitter du der med ett produkt som kan brukes, dog mangler det store deler, men det som ligger der kan brukes fullverdig. Så det er veldig positivt med å gjøre det på en sånn måte.”

Programmerar i hovudsak på tenar

Det har vore ein del diskusjonar internt hjå BU om korleis dei praktisk skal arbeide med iterasjonar. Eit punkt det har vore særleg mykje diskusjon rundt, er lengda på iterasjonane. BU har etter eige utsegn lagt seg på noko lengre iterasjonar, særleg i starten. Iterasjonane hadde ei lengd på opp i mot 4 månader, men har no vorte noko kortare. Argumentasjonen for så lange iterasjonar var at dei følte at det var ganske omfattande å starte opp ein iterasjon, og det er også omfattande å avslutte ein iterasjon, særleg i høve til testinga. Samstundes går lufta litt ut av folk ved ferdigstilling av ein iterasjon, og slik eg tolkar det, ønskte ein ikkje for mange slike ”pust i bakken” periodar. Men det er framleis ikkje full semje om kor lange iterasjonane bør vere.

”Jeg ser veldig mørkt på mindre iterasjoner, fordi det er så veldig intenst når en holder på, spesielt mot slutten.”

Systemutviklar med ansvar for analyse og i særleg grad design

”Når vi har iterasjoner på to til tre måneder så blir det vanskelig å få til effektivt arbeid første halvdel av iterasjonen. Hadde iterasjonene vært på to til tre uker, så hadde det sannsynligvis blitt mer effektivt.”

Systemutviklar med hovudvekt på teknologisida/programmering

Det er nok rett å seie at arbeidsintensiteten har blitt merkbart høgare når det nærmar seg iterasjonsslutt enn ved oppstart. Ved at ein når slike intensive toppar har det vist seg vanskeleg å komme raskt i gang med oppstart av ny iterasjon. Dette gjeld særleg for dei som sit på implementering/programmeringssida. Dei må ”vente” med å starte

sitt arbeid i ein ny iterasjon til analyse og design har utarbeidd use case dei skal realisere i iterasjonen. BU har freista å ta tak i dette ved at ein i avslutningsfasen av ein iterasjon tek tak i element som skal realiserast i starten av neste iterasjon. Dei håper med dette å redusere den potensielle dødtida dei har erfart for enkelte arbeidsflytar i oppstart av ny iterasjon.

Ein fare i forhold til å jobbe mot faste fristar er at ein i avslutningsfasen kan få problem med å halde tidsfristar. Dette kan medføre at arbeidet som vert levert, ikkje har den kvaliteten det burde ha. Det kan og resultere i at ein skuvar element over til neste eller seinare iterasjonar, og med dette utsett problemet. Ein kan då risikere å ende opp i ein vond sirkel der for mykje vert utsett frå iterasjon til iterasjon. Dette vil fort hope seg opp til uhandterlege mengder med element ein må ta att. Når dette er sagt, vil eg presisere at eg sit igjen med eit generelt inntrykk av at dei fleste tilsette synest iterasjonar er ei grei arbeidsform.

Eit sentralt poeng med iterasjonar, slik eg ser det, er deira gjentakande karakter. I datamaterialet eg har samla inn, kjem det fram at denne gjentakande karraktaren spelar ei rolle i forhold til kor hurtig prosjektmedlemmene er i stand til å ta i bruk metoden ut ifrå dei føresetnadane som bedrifta har lagt til grunn for arbeidsforma. Dette vil eg komme nærare tilbake til under punkt 7.4. For ein vidare diskusjon kring effektivitet og kvalitet knytt til iterasjonar sjå Pilskog sitt arbeid (Pilskog 2003)

7.1.3 Arbeidsflytar

Å tildele dei tilsette spesifikke roller og fordele dei i team i dei ulike arbeidsflytane, er eit av særtrekka ved RUP. For BU sin del har dei formelt definert og plassert tilsette i ulike roller, men grunna BU sin storleik, har fleire tilsette ofte måtte kle meir enn ei rolle. Rollene er altså formelt definert, men personane som kler dei overlappar kvarandre. Det kjem til uttrykk at denne rollefordelinga verka positiv i høve til arbeidsdelinga.

”... der er kanskje flere roller enn det vi har vært formelle på å definere tidligere. Så det føler jeg har hjulpet oss, med å fordele oppgavene på en fornuftig måte.”

Mellomleiar med hovudansvar for det tekniske

Dette var som tidlegare i rapporten nemnt første gong BU delte opp arbeidsstokken på ein slik måte. Sjølv tildelinga av roller var relativt uproblematisk. Her kom eit av særtrekka ved BU tilsyne. Dei fleste tilsette vart plassert etter eige ynskje og ikkje tildelt ei fast rolle frå leiinga.

Eit moment som har hatt negativ innverknad på systemutviklingsprosessen til BU er at store delar av kravsarbeidet til prosjektet vart gjennomført før dei valde å nytte RUP som systemutviklingsmetode. Fleire har gitt uttrykk for at krava har vorte statiske og vanskelege å endre. I tillegg har det vore eit problem at krava som heile ikkje er presise nok, og veldig rundt formulerte. Eksempel på dette er at ein i kravdokument finn formuleringar som: ”det stilles krav til grensesnitt”

”Våre krav har vel som dere har sett en ytterst rund formulering, og du kan legge hva du vil i de kravene”

Systemutviklar med ansvar for analyse og i særleg grad design

At krava er statiske og vage har medført at det gjennom arbeidsprosessen har vore nødvendig å endre og reformulere krava. Dette har medført fleire problem i dei tidlege iterasjonane, mykje fordi BU ikkje har hatt noko godt verkty for å spore endringar i krava. Kravhandteringa er noko dei i seinare iterasjonar har sett av tid til å utbetre.

Analysearbeidet har sjølvstakt vorte tyngre grunna dei nemnde problema knytt til krava. Eit interessant moment ved analyse- og designarbeidet til utviklingsgruppa har vore kvar skilje mellom design og analyse skal gå. Teoretisk sett er skiljet klårt definert i høve til kva som er analyse og kva som er design. Eit use case vert først analysert og så designa før det vert sendt vidare til implementering. Men personane som utfører dette arbeidet er i fleire høve dei same. Dermed vert skiljet noko uklårt. Dette er ikkje nødvendigvis negativt, men kan vere problematisk i høve til å presisere kvar eit use case er i utviklingsprosessen. I mitt datamateriale er ofte analyse- og designfasane omtalt som ein og same fase, analyse/design.

Eit mykje diskutert emne mellom design- og implementeringsteama er på kva detaljnivå designet skal ligge. Deltakarar i implementeringsfasen har vore dei som har uttrykt mest misnøye så langt. Det har kome til uttrykk at dei ikkje er nøgde med

detaljnivået i designa dei får oversendt. Dei har følt i dei tidlege fasane av prosjektet at RUP er ein metode som ikkje legg særleg til rette for implementering, og er for fokusert på analyse og design. Dette inntrykket meiner eg har endra seg noko i løpet av prosessen, men det var særleg gjeldande for dei første iterasjonane.

”Jeg har ikke inntrykk av at RUP sier så veldig mye om hvordan vi skal programmere der. Jeg tror den stopper etter design, og så legger den selvfølgelig styring på hva du skal gjøre hver iterasjon, og hvor lenge den skal vare.”

Programmerar i hovudsak på tenar

Testfasen er dei fleste enige om har fungert tilfredsstillande. Dei har i stor grad gjort nytte av automatiske testar i implementerings arbeidet. Det har vore krav til at all kode skal vere køyreleg før den vert lagt inn. I tillegg har ein henta inn sluttbrukarar i sluttfasane av kvar iterasjon og kjørt fulle systemtestar, der ein kontinuerleg rettar opp feil der det har vore mogleg.

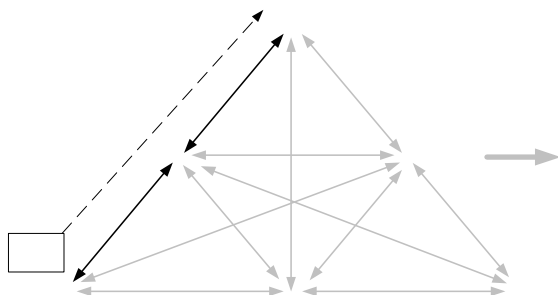
7.2 RUP verktøy eller regel

I analysedelen i denne rapporten presenterte eg ein aktivitetsteoretisk modell for BU sitt arbeid. I denne delen vart det kort diskutert kvar RUP skulle plasserast i Engestrøm sin figur for eit aktivitetssystem. Var det eit verktøy eller eit regelsett? Eg vil her ta opp den tråden og diskutere det nærare. I artikkelen ”Learning, Working and Imagining” (Engestrøm 1990) synleggjer Engestrøm korleis subjekt og dels samfunn relaterer seg til verktøya, og korleis denne relasjonen er i endring.

Då RUP først vart introdusert for prosjektgruppa var det ikkje utan ein viss skepsis blant dei tilsette. Fleire frykta at dette var ein metode som ville verke hemmande på effektiviteten, og legge for store føringar på korleis ein skulle arbeide. Slik eg ser det, er det riktig å seie at RUP for fleire vart oppfatta som eit sett reglar som ville styre arbeidssituasjonen deira. I høve til Engestrøm sitt triangel vil eg seie at RUP i dei første fasane av prosjektet må plasserast inn under regelnoden.

Etterkvart som prosjektet har skride fram og dei tilsette har vorte meir kjende og fortrulege med metoden, meiner eg å sjå at oppfattinga av RUP har endra seg. RUP har, etter mitt syn, vandra frå å vere eit sett reglar dei måtte relatere seg til, til å verte

eit verkty dei nyttar meir eller mindre medvite. Eg vil følgjeleg hevde at RUP har vandra langs den vertikale aksen til venstre i figur 11 frå regelsett til verkty.



Figur 11 RUP regel eller verkty?

Dette synet på RUP som eit verkty vart stadfesta av dei tilsette ved BU under ein presentasjon som Pilskog og eg hadde for alle tilsette ved BU i Bergen. Eg vil kome attende til korleis synet på RUP har endra seg gjennom prosjektet, og korleis dette heng saman med at dei tilsette har lært seg å nytte RUP som metode (jf. 7.4.2)

At dei fleste respondentane sine utsegner heller mot å sjå RUP som eit verkty meir enn eit sett reglar dei er tvinga til å arbeide innanfor, kan tolkast som ein styrke for RUP og eit bevis for at ein kan ha gjort eit riktig val av metode. Dette kan sjåast i forhold til Bødker sitt kriterium for kva som er eit godt verkty (jf. 2.6.2). Metoden eller strukturen den legg opp til, vert nytta meir eller mindre bevisst i forhold til å nå det gitte objektet.

Intervjuar: Men så, kanskje kan du si at den har godt i fra å vær ett sett med regler som du fikk trett over deg til å komme litt, skli litt bak, og bli bare noe verktøy som du bruker mer eller mindre bevisst.

Mellomleiar med hovudansvar for det tekniske: Ja det er helt riktig.”

Subjekt

Det kjem og fram hjå respondentar som har gjort seg opp ei meining om RUP, at dei gjerne ser for seg at ein i det neste prosjektet også nyttar RUP som systemutviklingsmetode.

RUP

Reglar

7.3 Kommunikasjon og dokumentasjon

Korleis dei ulike deltakarane i ei prosjektgruppe kommuniserer kan vere avgjerande for resultatet i eit prosjekt av Forv@lting sin storleik og karakter. Cockburn (Cockburn 2002) peikar på at korleis ein kommuniserer kan vere kritisk i høve til om eit utviklingsprosjekt vil lykkast. Som det har vore nemnt tidlegare, er BU ei bedrift med relativt opne kommunikasjonskanalar. Dette gjeld både mellom dei tilsette og frå leiinga til tilsette. Den daglege kommunikasjonen har ein uformell karakter og kan finne stad kvar som helst og når som helst. Av fastsette kommunikasjonsfora må nemnast dei faste måndagsmøta der alle i prosjektgruppa kjem saman og der kvart enkelt team presenterer kva dei har gjort, og kva dei vil gjere vidare. Av andre faste fora er sjølvstøtt oppstarts- og avslutningsfasane for den enkelte iterasjon sentrale.

Dei fleste eg har vore i kontakt med, trekker fram den munnlege og uformelle tonen i den daglege kommunikasjonen som noko positivt. Terskelen for å søke hjelp, støtte eller få utfyllande kunnskap knytt til ei spesifikk utfordring, er låg. Dei tilsette gjev uttrykk for at RUP i liten grad har endra kommunikasjonsmønstra i den daglege interaksjonen, men det må her nemnast at få tilsette hadde eit reflektert forhold til dette. Eit element som gjennom iterasjonane har endra seg, er "språket" ein nyttar. At RUP no er utviklingsmetode, legg føringar på kva termar ein nyttar, og ein er avhengig av å ha ei felles forståing av kva desse RUP- spesifikke termane inneber. Dette vil verte vidare diskutert under punkt 7.4.2

Som det tidlegare er nemnt (jf. 7.1.3) var det i dei tidlege fasane problem i forhold til kommunikasjon mellom analyse/design og implementeringssida. Dette hadde nok samanheng med at ein ikkje heilt hadde funne forma for kva som skulle overleverast frå design til implementering. Det var uklart kva detaljnivå modellar, use case og diagram skulle ha. Denne kommunikasjonen har vorte merkbar betre etterkvart som ein har nådd nye iterasjonar, og ein har funne si form for å kommunisere designet. Det må her og seiast at ein i større grad har teke med tekstleg beskriving av innhald i form av eit Word-dokument.

”Vi har slitt litt med at vi får ikke helt til den informasjonsflyten mellom oss og kodeguttene så godt som vi hadde ønsket.”

Systemutviklar med ansvar for analyse og i særleg grad design

” I begynnelsen hadde vi litt problemer med det at vi er en gruppering som jobbet med design og en gruppering som jobbet med teknologi og arkitektur. De jobbet fullstendig uavhengig, som førte til om ikke konfliktsituasjoner, så til litt synkroniseringsproblemer når vi kom et stykke ut i det. Og vi merker jo nå for hver iterasjon hvordan vi definerer grensene bedre”

Systemutvikler med hovedvekt på teknologisiden/programmering

Det er og påfallande i kommunikasjonen mellom desse fasane at begge teama meinte at den andre sida nytta for lang tid for å utføre sine arbeidsoppgåver. Særleg på implementeringssida har det kome til utrykk frustrasjon over å måtte vente på å få overlevert dei første use case i samband med iterasjonsoppstart.

Det vert trekt fram at use case og aktivitetsdiagram er ein god måte å kommunisere med brukarane på. Det hjelper til å avgrense problemområdet, noko som kan vere veldig viktig når ein skal diskutere løysingsforslag, at ein konsentrerer seg om mindre bitar av gongen. Modellane kan og nyttast på ein slik måte at personar som ikkje har teknisk spisskompetanse, skjønar kva det dreier seg om. Det vert poengtert at dette har vore til hjelp i fasane der ein skal spisse krava i analyse- og designfasen. Dei har og vore nyttige i kommunikasjonen med sluttbrukar i testfasane, når ein vil gjere endringar i høve til realiserte use case.

Eit anna viktig kommunikasjonsforum er dei ulike teama sine uformelle møte og diskusjonar. Det kom fram under intervju at deltakarane i dei ulike teama hadde relativt hyppige uformelle møte der dei i fellesskap fann løysingar på spesifikke problem eller utfordringar dei stod over for. Slike møte tok ofte form av at dei samla seg rundt ei tavle og skisserte ulike løysingar. Så diskuterte dei seg fram til den beste løysinga. Problemet med desse uformelle møta var at dei i liten grad dokumenterte den løysinga dei hadde vorte einige om. Når dei gjekk kvar til sitt etter ein slik diskusjon, hadde dei ein tendens til å ha si eiga personlege oppfatning av kva dei hadde vorte einige om.

Utviklar med ansvar for programmering på klienten: ... det å få fortalt det til alle, og også rett og slett det å få det ned på papiret, så du kan gå inn å se på det etter du har bestemt det. Blir litt sånn, du har gjerne en diskusjon en tre stykker, blir enige om noe, også glemmer vi jo alt annet sant.

Intervjuar: Får litt selektiv hukommelse når man går hver for seg?

Utviklar med ansvar for programmering på klienten: (latter) Ja.

Behovet for kanalar ein kan kommunisere endringar og forbetringar gjennom oppstår ikkje berre etter slike uformelle møtesekvensar. Det kom og fram at enkeltpersonar gjerne kunne tenkt seg faste kanalar for å distribuere sine forbetringar og tankar, og ikkje minst kunne finne fram i andre sine gode idear og løysingar.

”Viss du gjør designendringer, så en ting er å få oppdatert diagrammer og dokumentasjon og sånt vi har, Men det er jo også noe med å få formidlet den endringa til alle på prosjektet. Lager du smarte ting, gjør en refactoring kanskje og trekker ut og lager noe generell kode. Hvordan får vi formidlet best mulig til alle når den er tilgjengelig. Sånn at de på en måte kan finne ut av det uten å hver gang måtte lete gjennom all kode for å se om det skulle være noe her.”

Programmerar i hovudsak på tenar

Det vert frå fleire personar peika på at dei har ulike verkty som skal støtte denne kommunikasjonen og formidlinga, men dei vert på noverande tidspunkt ikkje utnytta. Dette behovet var særleg synleg i høve til foredlinga av krava. Krava var som nemnt tidlegare upresise og statiske. Desse krava måtte spissast i utviklingsprosessen for å verte nyttige. Eit problem som her oppstod, var at ein sleit med sporing i høve til kva krav var som var endra, når var dei endra og kven som endra dei. Det har her vore arbeidd med ein database som betre skal hjelpe til med å halde styr på kravendringane, men denne var under utarbeidinga av dette arbeidet ikkje ferdigstilt. Utviklingsgruppa disponerer og ein internweb som kan nyttast til eit felles kommunikasjonsforum, men denne har ikkje i skrivande stund vorte utnytta slik dei hadde planar om.

”Vi har en web der vi egentlig skal legge ut alt. Det er ikke alle som er like flinke til det, men der skal vi egentlig legge ut alt.”

Mellomleiar med hovudansvar for det tekniske

Eg trur at BU har eit stort uforløyst potensiale i forhold til å nytte datastøtta verkty til å handsame og ta vare på denne typen informasjon og kommunikasjon. Det kom fram på presentasjonen eg og Pilskog hadde for BU, at dette var ting dei arbeidde med. Men dei hadde enno ikkje funne heilt ut korleis dei skulle gjennomføre det. Men balansegangen her vil alltid vere hårfin. Det gjeld å finne balansen mellom for mykje og for lite dokumentasjon. Det er fleire faktorar som spelar inn i høve til korleis og kor mykje ein skal dokumentere. Det første er verdien av dokumentasjonen ein sit igjen med. Det er ikkje matnyttig å dokumentere alt ein gjer i ein slik utviklingsprosess. Ein vil rett nok sitte igjen med eit gjennomdokumentert produkt, men kva verdi vil alle desse dokumenta ha? Vil dei nokon gong verte nytta etter at dei har vorte utarbeidd? Det vil vere viktig å vere i stand til å skilje mellom kva dokumentasjon som vil vere nyttig både i høve til drifting, vedlikehald og vidareutvikling, og kva dokumentasjon som vil vere av meir triviell og unyttig karakter. Ein må og heile vegen vurdere tidsforbruk i dokumenteringsarbeidet i høve til verdien av dokumentasjonen.

Om ein freistar å ta i bruk ei form for verktystøtta formidling av dokumentasjonen, er det to kvalitetar ved eit slikt verkty som bør ha hovudfokus. Det første er at distribuering av dokumentasjon skjer på ein slik måte at medlemmene i prosjektgruppa får informasjon dei anten må ha, eller svært truleg vil ha nytte av. Om dei vert overlessa av generell informasjon som ikkje vedgår dei, vil ein slik kommunikasjonskanal bryte saman. Den relevante informasjonen vil forsvinne mellom alt som ikkje er interessant for den einskilde prosjektmedlem. Det andre er at dokumentasjonen må verte lagra på ein slik måte at det er lett å finne fram i den når ein søker etter informasjon. Det må eksempelvis vere slik at dei siste endringsforslaga er dei du får fram for det use case du søker informasjon om. Vidare må ein finne ein struktur og organisering som alle i prosjektgruppa er einige om og innforstått med. Om ein klarer å oppnå ei slik semje, har ein eit potensiale for å skape ikkje berre eit godt grunnlag for kommunikasjon medlemane imellom, men ein vil i større grad kunne spore endringar tilbake til person og tid. Eit anna positivt element er at det kan hjelpe i utarbeidinga av dokumentasjon til sluttbrukar. I tillegg kan ein skape ein kollektiv kunnskapsbase for bedrifta. Dette opnar for eit større potensiale i framtidige utviklingsprosjekt.

”Viss noen ved hjelp av verktøy støtte kunne ha sporet tilbake til ett brukermøte og detaljene der så hadde det vært, det kunne oppklart mye! ... sporbarhet av kommunikasjon er i hvert fall en mangelvare. Det blir veldig masse store Word dokumenter uten noe kryssreferanser.”

Systemutviklar med hovudvekt på teknologisida/programmering

Ein annan veg å gå i høve til dokumentasjon er Ekstrem Programmering (XP). Det kjem fram i fleire intervju, særleg hjå programmerarane, at dei finn tankar og moment ved XP interessante. Det vert hevda at dei allereie nyttar moment frå denne tankegangen, som automatiske einingstestar, før dei legg inn koden. Men det eg særleg tenker på i denne omgang, er at ein i XP gjerne skreller vekk mesteparten av den skriftlege dokumenteringa, og baserer seg i større grad på munnleg kommunikasjon - noko som kan vere effektivt og kreativt. Dokumentasjon vil berre verte utarbeidd om kunden krev det. Dette er noko eg trur kan fungere veldig godt i enkelte samanhengar, men i eit prosjekt av denne storleiken trur eg det er viktig å ha ein god og oppdatert dokumentasjon gjennom utviklinga. Men som eg nemnde over, vil det vere ein balansegang i forhold til verdien av dokumentasjonen og den tida ein investerer i han.

”Jeg er vel litt skeptisk til å gjøre relativt store ting bare på XP metoden for å si det sånn. Så det kan være greit å ha noe som tvinger deg til å gjøre litt analyse/design, lage en del dokumentasjon og opprettholde en del dokumentasjon, samtidig som du skreller vekk såpass mye at det ikke kveler deg.”

Programmerer i hovudsak på tenar

7.4 Læringseffektar

Eit sentralt punkt i forskinga som her vert presentert er om der finst spor eller bevis for at dei tilsette si læring endrar seg gjennom å arbeide med RUP. Eg vil her skilje mellom tre område for læring. Det er eine er læring knytt mot metoden. Er metoden lett å lære? Vil dei rutinerte systemutviklarane ha problem med å ta RUP i bruk, eller vil det skje tilnærma smertefritt? Det andre området for læring eg vil fokusere på, er læring knytt mot fagkunnskap innan domenet systemet skal nyttast i. Til slutt vil eg omtale det eg har valt å kalle teknisk læring. Med omgrepet teknisk læring vil eg fange opp læringseffektar knytt til tekniske aspekt som UML, objektorientering og andre teknologiar nytta i prosessen.

Som nemnt tidlegare i rapporten, har det vore mykje opp til mine respondentar å definere og beskrive kva, om noko, dei har lært så langt i arbeidsprosessen. Eg var frå byrjinga klar over at dette kunne vise seg å vere vanskeleg, og det vart det. I dei fleste intervjusituasjonane hadde ikkje respondenten på førehand noko klart og reflektert forhold til eiga læring. Og dei hadde då følgjeleg ikkje før intervjusituasjon gjort seg opp klare tankar om kor vidt læringsprosessen så langt skilde seg frå tidlegare arbeid.

Arbeidet med finne ut kva faktorar ved RUP som påverkar læring, har difor vore tidvis vanskeleg sidan svara ikkje har vore klåre og eintydige, men heile tida må sjåast i forhold til andre utsegn respondenten kjem med. Og ut ifrå heilskapen i det som vert sagt, kan ein seie noko om kva læringsprosess respondenten har vore gjennom og korleis den har vore i høve til tidlegare arbeid.

7.4.1 Domenekunnskap

Systemet under utvikling er knytt mot eit spesifikt felt innan norsk offentleg forvaltning. Fagfeltet er av relativt stort omfang, og som innan dei fleste større felt eller sektorar er det eit stort sett spesifikke lover, reglar og andre vedtekter ein må ta omsyn til. Det er kritisk viktig at systemet ikkje inneheld formelle faglege feil, då desse kan få store konsekvensar både for enkeltpersonar og organisasjonar.

Det må vidare nemnast at fleire av dei tilsette har godt kjennskap til fagfeltet, då BU har drive forvaltningsarbeid innan denne sektoren over lengre tid. Så det var særleg mellom respondentar med kort fartstid i BU eg fann verkeleg klare tendensar i retning av læringseffektar i forhold til det eg har valt å kalle domenekunnskap. Men eg fann og støtte for funna mine i intervjurespondentar med lang fartstid.

Mest alle respondentar som sat på implementeringsdelen av systemutviklinga, gav uttrykk for at modellar, dokument og andre former for spesifikasjonar dei fekk overlevert frå analyse/design ikkje var av ein slik karakter at dei kunne programmere direkte ut i frå dei. I byrjinga fekk dei dette til å framstå som noko

negativt. Det vart og hevda at RUP ikkje er ein metode som ikkje tek særleg omsyn til arbeidet som skal gjerast etter analyse og design.

”Akkurat de use case diagrammene og aktivitetsdiagrammene føler eg at ikke er så veldig nyttige for meg som skal programmere, fordi de er jo ikke så detaljerte at du kan programmere alt etter de.”

Programmerar i hovudsak på tenaren

Men ettersom dette vart diskutert, kom det fram at det ikkje nødvendigvis berre var negativt. Det kom fram at med utgangspunkt i at dokumentasjonen var av ein såpass vag karakter vart det nødvendig for den einskilde programmerar å fullt ut forstå kva det var han skulle implementere. Dette kan sjåast på som ei rake motsetning til det som på BU vert omtalt som lefseprogrammering der ein av praksisane er at ein får overlevert eit sett av pseudokodar som skal realiserast.

”... så får en nå mer eller mindre et sett med vage diagrammer og en del use case og sånt. Så en må forstå problemområdet og det blir en god del dialog og diskusjon med andre som jobber med samme tingen for å forstå hva en egentlig skal lage... Og det tvinger deg til å forstå fagområdet du jobber med”

Systemutvikler med hovedvekt på teknologisiden/programmering

”Sånn som vi benytter det så er programmeringsgrunnlaget mer abstrakt en det vi har hatt før, så det tvinger frem en læringsprosess for fagområdet, men det tvinger deg også selvfølgelig at du lærer RUP notasjonen og sånt...”

Systemutviklar med hovudvekt på teknologisida/programmering

Det var ikkje berre tilsette på implementeringssida som drog fram at dei følte at dei hadde fått betre overblikk over og innsyn i fagfeltet gjennom dette prosjektet. Også personar som arbeidde på analyse- og designsida fann at RUP hadde positiv innverknad på læringa i høve til fagområdet. Iterasjonane har også her spelt ei liknande rolle. Men det kom og fram fleire positive trekk ved iterasjonar knytt mot læring. RUP sine krav knytt til iterasjonar og kommunikasjon mellom dei ulike teama gir den einskilde tilsett eit betre oversyn over korleis prosjektet ligg an og kva faglege problem som er knytt mot dei enkelte use case.

”Det har stilt meg i en posisjon der jeg har fått fagkunnskaper på dette området. Hadde jeg vært f.eks. lefseprogrammerer på dette prosjektet, hadde jeg ikke kunnet(fagfeltet) så godt som i dag. Det at jeg har fått jobbet så mye med hele fagområdet, det har nok brakt meg langt videre, spesielt fagområdet vil jeg si.”

Systemutviklar med ansvar for analyse og i særleg grad design

Det vart og poengtert at RUP gjev utviklarane ei meir heilskapleg forståing av prosjektet ein arbeider på. Respondentar poengterte at utviklinga var mindre segmentert enn tidlegare prosjekt dei hadde delteke i. Gjennom dette fekk ein eit betre overblikk over heilskapen. Eit anna positivt trekk ved RUP var at overgangen frå å arbeide mot brukarane til å skulle realisere prosjektet har fungert på ein naturleg måte.

”Det har vært en naturlig gli å gå i fra brukerne og inn i vår tekniske verden, og det er klart det har vært veldig lærerikt både for det faglige og det tekniske...”
Systemutviklar med ansvar for analyse og i særleg grad design.

7.4.2 Metodekunnskap

Gjennom analysane av intervjua har eg og kome fram til at dei tilsette tek metoden relativt hurtig til seg. Sidan eg ikkje har noko å samanlikne med, har eg ikkje dekning for å hevde at ein lærer RUP raskare enn andre metodar. Men eg meiner å ha dekning for å påstå at RUP er ein metode nye brukarar lærer hurtig. Dette kan knytast til særtrekk ved metoden. Eg vil særleg trekke fram iterasjonane som hovudgrunn for dette. Fleire av intervjuobjekta drog fram dette som eit positivt moment ved RUP.

”... , men det du ser det å repetere ting, det er deilig rett og slett fordi du gjør ting bedre neste gang. Så føler du at du gjør det enda bedre gangen etter der igjen. Fordi ting blir kjent. Så det er rett og slett en form for trening, og det, vi må læres dette først og så går det mer eller mindre av seg selv etter hvert.”
Mellomleder, med hovud ansvar for det tekniske.

RUP sin repeterande karakter med utspring i iterasjonane er noko som har vorte poengtert fleire gonger i dette arbeidet. Men i forhold til kor raskt metoden vert innprenta og fell naturleg for prosjektgruppa sine medlemmer, er dette etter mitt syn særskilt sentralt.

”Det at en får flere forsøk. Og det felles språket vårt, hvordan vi uttrykker oss, vil jo konvergere mot at vi alle vet hva vi snakker om.”
Konsulent med ansvar for analyse og design. Hovedfelt design

I oppstartfasen av prosjektet var mange utrygge med tanke på kva det var ein hadde gjeve seg ut på. Hadde ein valt riktig metode for prosjektet, eller var dette ein metode som var for stor og altomfattande og ville kvele heile prosjektet? Dei første fasane i prosjektet var heller ikkje uproblematiske. Det var mellom anna ein del friksjon mellom design og implementering i høve til korleis ein skulle kommunisere i forhold til use case- realiseringa. Dette betra seg, som omtalt over, etter nokre iterasjonar når ein hadde fått på plass det nye språket ein nytta for å kommunisere.

”... , altså du lærer hva slags situasjon du vil være i. Du får ikke de store overraskelsene på slutten som du kanskje hadde fått dersom du hadde kjørt en stor iterasjon. Så det er jo absolutt positivt.”

Systemutvikler med hovedvekt på teknologisiden/programmering

Det at prosjektgruppa sine medlemmer har teke til seg metoden såpass raskt, kan også henge saman med at dei no vert nøyddde til å snakke same språk(jf. 7.3). Dette var ikkje berre uproblematisk i dei første iterasjonane. Men etter kvart som dei fann ei form med passende abstraksjonsnivå og detaljrikdom og alle vert fortruleg med dette, har metoden sett seg - også i høve til informasjonsflyten teama i mellom.

Ein drivande element i at ein jobbar mot eit felles språk, er at kvar enkelt vil vere avhengig av å heve sitt eige kunnskapsnivå for å kunne delta på ein fullverdig måte i kommunikasjonen knytt mot systemutviklinga. Om eit medlem ikkje tek seg tid til eller evnar å sette seg inn i det nye språket, vil han få problem både med å yte og forstå kva som skjer. Ein kan her seie at ein vert motivert ut ifrå eit behov for å kunne yte både sosialt og profesjonelt i det daglege arbeidet i utviklingsgruppa.

7.4.3 Teknologisk kunnskap

I Forv@lting-prosjektet har BU teke i bruk ei rekke nye teknologiske element. I tillegg til å anvende RUP som metode, er det første gongen dei freistar å gjer nytte av objektorientert programmering.

Tidlegare i dette kapittelet har eg trekt fram korleis dei tilsette har vorte tvinga til å endre språket, måten dei kommuniserer på. Eg vil her gå meir i detalj og trekke fram dei mest sentrale elementa i dette språket. Det første er at dei no nyttar UML når dei

skal beskrive krav og løysingsforslag. Å arbeide etter eit fastsett definisjonsspråk med ein fastsett notasjon gjer at når ein når ei felles forståing av kva den inneheld og finn sin måte å nytte den på, sikrar dette kvaliteten på informasjonen som vert sendt mellom dei ulike deltakarane i prosjektgruppa.

Ein av dei mest sentrale komponentane dei har nytta frå UML, er sjølvstapt use case. Som nemnt tidlegare har det vore ein del intern diskusjon kring korleis desse skal nyttast. Eg tenkjer her særskilt på på kva detaljnivå desse skal ligge. Use case har i oversendinga frå design til implementering vorte supplert med særleg sekvens- og aktivitetsdiagram og også ein del klassediagram. Men det må seiast at ein ikkje fullt ut enno har nådd full semje om korleis desse skal vere, sidan dei har funne det naudsynt å supplere kvart use case med eit Word dokument som også gjev ei tekstleg skildring av krava og designet til det gitte use case.

Det kjem og i ulike intervju fram at ein ser klare fordelar ved å strukturere arbeidet mot tekniske standardar for kommunikasjon. Ein ser og i større grad verdien av å ha utarbeidd use casea i tandem med arkitekturen.

”Det gjelder også arkitektur ting som gjelder rent sånt domene saker og sånn mer tekniske ting, de er lettere å se når du sitter og jobber med diagrammene og fått løftet blikket litegrann.”

Konsulent med ansvar for analyse og design

Det er og første gang BU nyttar objektorienterte programmeringstankar på eit heilt prosjekt. Ikkje alle var heilt fortruleg med å arbeide på denne måten i oppstarten, og ein såg heller ikkje gevinsten i tidlege fasar. Men etter kvart som prosjektet har skride fram, har fleire sett nytteverdien av det. Innan fagfeltet er det fleire element som er samanfallande i dei enkelte use case. Så langt dei no har kome i prosessen, har dei no kunna starte gjenbruk av allereie implementerte element. Slik gjenbruk av ulik kode og klassar viser at dei har greidd å utnytte potensialet som ligg i objektorientert utvikling. Dette er noko som må kunne sjåast på som eit bevis for at ein læringsprosess har funne stad hjå tilsette som frå før ikkje hadde særleg forkunnskap knytt til objektorientering. Kor vidt dette kan knytast til RUP, er det meir vanskeleg å seie noko konkret om utifrå mitt datamaterielle. Men eg vil hevde at det er truleg at elementa knytt mot læringseffektane i forhold til domene og

metode er samanfallande med læringseffektane for objekt orientering, då dette er ein integrert del av RUP som arbeidsmetode.

Gjennom å nytte RUP som systemutviklingsmetode har det og kome fram at kvart enkelt medlem av prosjektgruppa får eit betre oversyn over prosjektet som heile. Gjennom presentasjonane knytt mot iterasjonsslutt og dei faste måndagsmøta vert alle oppdatert ikkje berre på kva deira team driv med til dagleg, men og kva dei andre teama driv med, og kvar dei er i sine arbeidsprosessar. Samla gjer dette at alle til ei kvar tid veit kvar systemet som heile er i utviklingsprosessen.

7.5 Oppsummering

Under følgjer ei poengtert og kortfatta samling av dei funna eg har gjort, og som eg meiner er særleg relevante.

RUP må tilpassast prosjektet

RUP som metode er ikkje uavgrensa skikka til å kunne nyttast på alle prosjekt uavhengig av storleik og kompleksitet. I Forv@lting har utviklingsgruppa lagt prosesstankegangen til grunn og valt ut dei komponentane ved metoden dei meiner er passande for deira prosjekt. Eg trur dette har vore avgjerande i høve til å få metoden til å fungere såpass godt som den har gjort

Iterasjonslengde

Iterasjonslengda bør ikkje overskride tre til fire månader, då dette kan verke negativt inn på arbeidsintensiteten.

Arbeidsflytar og roller

RUP si inndeling av arbeidstokken i ulike arbeidsflytar og team har vore til hjelp med å fordele arbeidsoppgåvene for systemutviklingsgruppa.

Gjennomføre metoden frå start til slutt.

At kravinnsamlinga tok til før metoden var fastsett, har verka negativt inn på utviklingsprosessen. Eg trur dei første iterasjonane ville ha vore mindre problemfylte om RUP var fastsett som metode heilt frå oppstartsfasen av Forv@lting.

RUP frå regel til verkty.

Dei tilsette sitt syn på RUP har endra seg gjennom systemutviklingsprosessen frå å vere eit sett reglar ein skulle arbeide etter, til å bli eit verkty. RUP med sine metodar og strukturar vert nytta av dei tilsette meir eller mindre medvite for å nå det kollektive målet om ferdigstilling av Forv@lting.

Kommunikasjonsmønster og innhald har endra seg

BU har halde fast på dei munnlege og uformelle kommunikasjonsmønstra, men har teke i bruk RUP sin semantikk. Dei har også gradvis fått eit felles språk med høveleg abstraksjonsnivå, basert på UML, som dei kommuniserer utviklinga gjennom.

Dei har i tillegg utvida møtefrekvens med faste måndagsmøte som saman med samlingar ved iterasjonsstart og slutt gjer det enkelte prosjektmedlem meir medvite i forhold til utviklinga, og gir ei betre forståing av prosjektets progresjon og tilstand.

Dei har eit uforløyst potensiale i forhold til å nytte eit verkty for å støtte opp om dokumentasjon og endringskontroll. Tankar og planar knytt til dette er under utvikling. Det må likevel strekast under at denne dokumenteringa kan gå utover effektiviteten om ho vert for omfattande, noko som ikkje er ynskjeleg.

Ein lærer fagfeltet betre å kjenne gjennom RUP

Denne påstanden er særleg treffande for programmerarane som under realisering av ulike use case vert tvinga i større grad enn tidlegare å forstå innhaldet og implikasjonane for kvart use case. Detaljnivået i designet er ikkje på eit slik nivå at ein kan realisere dei direkte. Faste måndagsmøte, samt iterasjonsstart og avslutting spelar og inn på forståinga og kjennskapen til fagfeltet.

RUP er ein metode ein lærer hurtig.

RUP sin repeterande karakter med utspring i iterasjonane gjer at deltakarane i prosjektgruppa raskt tek til seg metoden. At ein får fleire moglegheiter og repeterer metodens innhald gjennom iterasjonane, vert av fleire peika på som positivt i høve til å få metoden til å sitte. Eit anna element som er medverkande er at dei vert tvinga til å bruke eit felles språk.

Teknisk Læring

Gjennom over nemnde punkt går det og føre seg ei teknologisk læring i form av at ein lærer å nytte ulike teknikkar knytt mot UML og dei ulike diagramma ein nyttar i utviklingsprosessen.

I Forv@ltingsprosjektet har objektorientert utvikling vorte nytta for første gang. I dei seinare iterasjonar kjem det fram at ein har byrja å sjå prov på at ein har nytte av dette gjennom gjenbruk av klassar og kode frå tidlegare realiserte use case. Objektorientering er ein integrert del av RUP. Men eg har i mitt datamateriale ikkje funne direkte prov på at prosesstankegangen og arbeidsmåtar i RUP har hatt innverknad på korleis dei tilsette har tilegna seg kompetanse knytt mot objektorientering. Men det er truleg at momenta nemnt i forhold til dei andre læringsområda er samanfallande.

8 Konklusjon og oppsummering

Til slutt vil eg vurdere resultata og drøfte nokre tankar kring dei funna eg har gjort i dette forskingsarbeidet. I tillegg vil eg diskutere kva verdi studien har, og i kor stor grad ein kan generalisere utifrå mine funn. Til siste vil eg skissere nokre potensielle vegar vidare ein kan gå innan det segmentet av fagfeltet eg har arbeidd med.

Ved inngangen til dette arbeidet hadde eg ei open haldning til korleis dette arbeidet skulle skride fram. Eg var svært spent på korleis vi ville verte mottekne hjå BU, i kva grad eg ville vere i stand til å stille dei rette spørsmåla, og om respondentane våre ville gje oss relevante svar. Eg var og spent på korleis aktivitetsteori ville fungere som rammeverk for studiet.

Arbeidet mitt har først og fremst vore retta mot kommunikasjon og læringseffektar. Men gjennom arbeidet dukka det sjølvstøtt opp andre interessante moment eg har funne det verdt å ta med i denne rapporten. Eg tenker her særleg på behovet for å tilpasse metoden til prosjektet, ikkje tilpasse prosjektet til metoden. Noko anna eg fann interessant er korleis personar med ulike stillingar hadde ulike syn på same sak. Det kom mellom anna fram at ein person som satt med ansvar for tenardelen meinte at designet dei fekk overlevert nok passa best for klienten, medan ein som sat med klientdelen meinte designet var mest tilpassa tenaren. Det var og interessant å sjå korleis kulturen som sit i veggane spelar inn i høve til korleis metoden vert teken i bruk. Eg meiner det gode arbeidsmiljøet på BU har hatt stor positiv innverknad i samband med innføringa av den nye metoden.

Når det gjeld hovudfeltet mitt, meiner eg at eg har gjort fleire interessante funn. Eg meiner det er klart å sjå at både kommunikasjonsmønstra og innhaldet har endra seg etter innføring av RUP som systemutviklingsmetode. Innhaldet eller språket dei kommuniserer med, har og endra seg mykje. I byrjinga hadde dei særleg mellom design og implementering problem med å ta i bruk dette nye språket på ein måte som begge grupper var nøgde med. Men gjennom felles innsats har dei etterkvart nådd eit felles språk som dei kan kommunisere med i systemutviklingsprosessen. I

starten av dette arbeidet var eg noko usikker på korleis eg skulle finne og påvise læringseffektar, som direkte kunne tilskrivast RUP, og ikkje minst føre prov for dette. Men allereie i dei første intervjusituasjonane kom det fram interessante forhold knytt mot læringseffektar som vart stadfesta og forsterka gjennom seinare sesjonar.

Å nytte aktivitetsteori som rammeverk har vore nyttig gjennom arbeidsprosessen. Aktivitetsteori har medverka til å fokusere på utviklingsprosessen som ein heilskap der fleire moment spelar inn, ikkje berre som ein individuell aktivitet for den einskilde utviklar. Det er viktig å ha blick for heilskapen, heile samfunnet og kulturen som er aktiv i og omkapslar utviklingsaktiviteten. Gjennom å nytte aktivitetsteori innan eit kvalitativt rammeverk, meiner eg at ein kan finne og adressere kulturelle og menneskelege sider ved utviklinga ein vanskeleg kunne ha funne på annan måte. Det er her sentralt å peike på at ein kan vanskeleg studere eit sosialt fenomen, i dette høvet RUP, utan å ha blick for kulturen det opptreir i.

Eit kritisk spørsmål ein kan reise i høve til dette, er om studiet som er gjennomført berre er gyldig for den kulturen eller det miljøet ein har studert. Spørsmålet om kor generaliserbare resultata er, er eit sentralt spørsmål i all forskning. Eg vil vere varsam med å hevde at alt eg har peika på, og funna eg har gjort, er universelle og vil gjelde for alle som vel å nytte RUP som systemutviklingsprosess. Likefullt vil eg hevde at dei gir gode peikepinnar på sentrale moment ved RUP som ein bør vere merksam på, om ein skulle velje RUP som metode. Eg vil her også vise til Pilskog (Pilskog 2003) sitt studium som har fokusert på effektivitet og kvalitet ved systemutviklingsprosessen.

8.1 Kva kunne vore gjort annleis

I ettertid er eg relativt nøgd med dei vala eg har gjort knytt mot metodologi og rammeverk. Eg meiner at gjennom å angripe problemstillinga med ein kvalitativ casestudie har eg fått innsikt og kunnskapar om BU og RUP eg ikkje ville fått gjennom eit klassisk kvantitativt opplegg basert på gjennomstrukturerte intervju, og forsøk på kvantifisering av funn og observasjonar. Fleire av dei mest interessante

observasjonane kom fram gjennom ein naturleg samtale om problemområde i intervjujesjonane, der informantane snakka fritt om det dei meinte var interessant og opptok dei. Rammeverket har, som eg nemnde over, vore til hjelp for å halde fokus på heilskapen i systemutviklinga.

kapittel

Eit spørsmål eg har stilt i ettertid er om eg burde ha gjort meg nytte av tilbodet om å tilbringe meir tid hjå BU. Som nemnt tidlegare var grunngevinga for valet eg gjorde eit ønske om å halde ein kritisk distanse til forskingsobjektet. Det må samstundes understrekast at eg har brukt ein god del tid hjå BU, også i uformelle samanhengar. Ved å ha vore endå meir aktivt tilstades, kunne eg nok avdekt meir kring bedrifta sine rutinar og arbeidsforhold. Det kunne ha vore særleg interessant å vere tilstades som passiv observatør under eit eller fleire av måndagsmøta deira, eller ved ein iterasjonsstart eller avslutting. Det kunne og ha vore interessant å ha delteke på eit av dei uformelle møta dei ulike teama held, men dette ville nok vore vanskeleg å gjennomføre. At eg var tilstades på eit slikt møte, ville nok gjort situasjonen kunstig og unaturleg for dei som var tilstades.

8.2 Vidare arbeid

Eit naturleg vidareføring vil vere å gjere ein liknande studie i anna bedrift som nyttar RUP for første gong, for å sjå om ein fann liknande trekk knytt mot læring og kommunikasjon, som kunne underbygge funna i denne studien. Eit anna interessant prosjekt vil vere å studere ein annan, gjerne agil metode, og sjå på skilnader når det gjeld kommunikasjon og læring metodane i mellom.

Eg trur og det kan vere interessant å gjere eit arbeid knytt mot dokumentering i og av systemutviklingsprosjekt. Ei tilnærming kan vere å setje RUP, som er ein relativt gjennomdokumentert prosess opp mot XP, som i liten grad legg opp til utarbeiding av dokumentasjon i sjølve utviklingsprosessen. Her vil det være særleg interessant å sjå på verdien av dokumentering knytt mot kvalitet og effektivitet i sjølve utviklingsprosessen.

Ein samanliknande studie av ulike systemutviklingsmetodologiar med fokus på sluttbrukarar og kundar si oppleving av ulike utviklingsprosessar, kunne og vore ein

interessant veg å gå. Eg tenkjer her særleg på korleis kommunikasjonen er mellom utviklingsteam og kunde, både i forkant, under utvikling og ved ferdigstilling. Kva føringar legg dei ulike metodane på relasjonane mellom kunde og utviklingsteam? Og kva innverknad har dette på det endelege produktet?

Moglegheitene for vidare forskning innan fagfeltet er mange. Forskinga som er presentert i denne rapporten er mitt bidrag til fagfeltet, og kan vere eit utgangspunkt for andre å bygge vidare på. Gjennom dette studiet har eg påvist sider ved RUP som ikkje nødvendigvis er opplagde og synlege ved første augekast. Arbeidet er basert på tidlegare presenterte studiar, tankar og idear. Arbeidet kan kanskje ikkje presentere lovmessige tilhøve som er universelt brukande. Likefullt meiner eg å ha bidratt med eit arbeid som er med å utvide forståinga av implikasjonar RUP kan ha for dei som vel å nytte den som systemutviklingsmetode.

Referansar

Barthelmess, P., & Anderson, K.M. (2000), *A View of Software Environments Based on Activity Theory*. Kluwer Academic Publishers.

Bødker, S. (1996). *Applying activity theory to video analysis: How to make sense of video data*. In B.A. Nardi (Ed.) *Context and Consciousness: Activity Theory and Human-computer Interaction*, 147-174. Cambridge, MA: MIT Press.

Cockburn, A (2002). *Agile Software Development*. Addison-Wesley

Dubé, L. & Robey, D. (1999). Software stories: three cultural perspectives on the organizational practices of software development. *Accounting Management and Information Technologies*, 9 (1999) 223-259.

Engeström, Y. (1990). *Learning, Working and Imagining*. Twelve Studies in Activity Theory. Orienta-Konsultit Oy Helsinki.

Engeström, Y. (1999). *Expansive Visibilization of Work: An Activity Theoretical Perspective*. In *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 63-93. Kluwer Academic Publishers.

Fitzgerald, B., *An Empirical Investigation into the adoption of systems development methodologies*. (1998). *Information Management*, Vol. 34, No. 6, s. 317-328.

Jacobson, Booch, Rumbaugh (1999). *The Unified Software Development Process*. Addison-Wesley

Jordan, B. and Henderson, A. (1995). *Interaction Analysis: Foundations and Practice*. *The Journal of Learning Sciences* 4 (1), 39-103

Kaptelinin, V., Nardi, B.(1997). *Activity Theory: Basic Concepts and Applications*.
In <http://www.acm.org/sigs/sigchi/chi97/proceedings/tutorial/bn.htm#U9>.

Koschmann, T. (1996). *CSCL: Theory and Practice of an Emerging Paradigm*.
Mahwah, NJ: Lawrence Erlbaum Associates

Kuutti, K. (1996). *Activity Theory as a Potential Framework for Human-Computer Interaction Research*. In B.A. Nardi (Ed.) *Context and Consciousness: Activity Theory and Human-computer Interaction*, 17-44. Cambridge, MA: MIT Press.

Kvale, Steinar (2001). *Det Kvalitative Forskningsintervju*. Gyldendal Norsk Forlag.

Martin (1982, 1992) I Dubé, L. & Robey, D. (1999). Software stories: three cultural perspectives on the organizational practices of software development. *Accounting Management and Information Technologies*, 9 (1999) 223-259.

Pilskog, Espen (2003) Hovudfagsoppgåve *RUP I praksis – En studie av positive og negative aspekter mht. produktivitet og produktkvalitet*. Universitetet i Bergen

Remenyi, D., Williams, B., Money, A., Swartz, E. (1998). *Doing Research in Business and Management. An Introduction to Process and Method*. Sage Publications.

Seaman, Carolyn B. (1998). *Qualitative Methods in Empirical Studies of Software Engineering*. *IEEE Transactions on Software Engineering*, Vol. 25, no.4, July/August 1999.

Sharp, H., Woodman, M., Hovenden, F., Robinson H. (1999). *The role of "culture" in software process improvement*. Institute of Electrical and Electronics Engineers, Inc

Säljö, Roger (2000). *Læring i praksis –Et sosiokulturelt perspektiv*. Cappelen Akademiske Forlag.

Walsham, G., (1995). *Interpretive Case Studies in IS Research: nature and method*.
Department of Management Science, The Management School, Lancaster
University.

Web referansar

Rational Software Company

<http://www.Rational.com>

Object Management Group

<http://www.omg.org>

Centre for CHAT & DWR

<http://www.edu.helsinki.fi/activity/pages/chatanddwr/activitysystem/>

Informal Education

<http://www.infed.org>

<http://www.infed.org/biblio/learning-social.htm>

Figurliste

FIGUR 1 RATIONAL UNIFIED PROCESS SI HISTORISKE UTVIKLING (JACOBSEN ET AL. 1999).....	8
FIGUR 2 ARBEIDSFLYTAR I EIN ITERASJON (JACOBSEN ET AL. 1999)	13
FIGUR 3 UNIFIED PROCESS SIN LIVSSYKLUS (RATIONAL.COM).....	15
FIGUR 4 STRUKTUREN AV AKTIVITET PÅ TRE NIVÅ SETT FRAM AV LEONT'EV (CENTRE FOR CHAT & DWR).....	22
FIGUR 5 ENGESTRØMS AKTIVITETSSYSTEM (ENGESTRØM 1990)	23
FIGUR 6 DUBÉ OG ROBEY SI FRAMSTILLING AV EIGEN METODE (DUBÉ OG ROBEY 1999).....	31
FIGUR 7 COCKBURN SIN "ELEMENTS OF A METHODOLOGY"(COCKBURN 2002).....	34
FIGUR 8 FORSKINGSDESIGN FRITT ETTER DUBÉ OG ROBEY (DUBÉ OG ROBEY 1999).....	38
FIGUR 9 SOUND CAPTURE SITT GRENSESNIITT MED AUTENTISKE LYDINSTILLINGAR	43
FIGUR 10 TRANSANA SITT GRENSESNIITT MED DØME PÅ EIT TRANSKRIPT OG LYDSTRAUM.	44
FIGUR 11 RUP REGEL ELLER VERKTY?	71

Vedlegg 1: Generell intervjuguide

1. **Hva er dine arbeidsoppgaver? (eventuelle roller)**
 - I BU generelt?
 - I prosjektet?

2. **Hvor lenge har du jobbet i BU?**
 - Flere forskjellige stillinger?

3. **Hvordan er det å jobbe i BU?**
 - Kontra andre jobber du har hatt?
 - Et sted du kan være lenge?
 - Utfordrende arbeidsoppgaver?
 - En god systemutviklingskultur?
 - Arbeidsmiljøet? Blir det gjort noe for å få et godt arbeidsmiljø?

4. **Hvilken tidligere erfaring har du?**
 - Utdanning?
 - Annen IT-erfaring?
 - Tidl. erfaring med andre metodologier enn RUP?
 - Tidl. erfaring med UML?

5. **Hvilke andre utviklingsmetoder enn RUP var aktuelle og hvorfor ble denne valgt?**
 - Ble de ansatte informert om utvelgingsprosessen?
 - Hvem bestemte dette?
 - Hvordan ble RUP introdusert?
 - Kurs?
 - Dine kurs?
 - Læringsutbytte av kursene mht. ressurser investert..
 - På hvilken måte sikrer BU at de ansatte har den kompetanse som trengs for å gjennomføre arbeidsoppgavene sine?

6. **Hvordan var dine tanker om objektorienterte utviklingsmetoder (OO) før prosjektets oppstart?**
 - Hadde du noen tanker i det hele tatt?
 - Preferanser til andre utviklingsmetoder?
 - Prosjektet vs. OO. OO positivt eller negativt for prosjektet?

- 7. Hva var hovedutfordringen med den nye arbeidsmetoden?**
- Så du noen problemer ved slike utviklingsmetoder?
 - Use-case drevet
 - Inkrementell
 - Iterativ
 - Arkitektur drevet
- 8. Arbeidsdeling: Hvordan følte du fordeling av oppgaver og ansvar innen organisasjonen fungerte i prosjektet?(dette er en av hovedmålsetningene for UP)**
- Følte du at rammeverket (RUP) definerer de forskjellige aktivitetene, rollene og leveransene (produktene) som skulle inngå i prosjektet på en tilfredsstillende måte? Hvorfor/hvorfor ikke?
 - Hvem bestemte arbeidsdelingen?
 - Etter personlighet?
- 9. Hva var ditt personlige mål ved inngangen til prosjektet?**
- Endret dette seg i løpet av prosjektet? Hvorfor?
- 10. Hvilken rolle føler du at RUP har spilt i prosjektet for deg personlig?**
- Verktøy eller regler? (Jmfr. Engeström artikkel) (hvordan forholder subjekt og samfunn seg til verktøyet?)
 - Generell kommentar?
- 11. Hvor mye teller det rent menneskelige kontra det tekniske/teoretiske i RUP i praksis? ("human factors in the software development process" – Gittens et. Al)**
- Menneske vs. Rammeverk. Hva teller mest for prosjektsuksess?
 - Hvilke av de to spiller mest inn for produktets kvalitet etter din mening?
 - Hvilke av de to spiller mest inn for gruppens produktivitet etter din mening?
- 12. Inneholder RUP en passe mengde regler osv etter din mening?**
- Hvorfor/ hvorfor ikke?
- 13. Hva kan du si om regler (eksplisitte og implisitte normer, sosiale relasjoner) utenom RUP sine arbeidsinstrukser i BU?**
- Finnes det noen uskrevne regler som spiller inn på produktet?
 - Interne arbeidsinstrukser?

- 14. Hvem bestemte kravene i prosjektet og hvordan ble de innhentet?**
- Fra hvilke sluttbrukere?
 - Mye fra BU internt
 - Departementet?
 - Hvem fra BU ”hentet de ut”?
 - Fungerte dette bra?
 - Forskjeller på kravuthentingene her kontra andre prosjekter?
- 15. Føler du at systemets verdi for brukerne var i fokus under utviklingen?**
- 16. Eksterne påvirkningskrefter/Påvirkning fra samfunnet:
Hvordan har eksterne parter (eks. politikere, andre sluttbrukere) påvirket systemutviklingen?**
- Satte bare rammene til å begynne med?
 - Aktive gjennom hele prosessen?
 - Forstyrrende eller til hjelp mht produktets kvalitet?
 - Hvem føler du har hatt ”reell” innflytelse på BUs arbeid og derav prosjektet?
- 17. Hva kan du si om bruk av verktøy i utviklingen?**
- Hvem bestemte hvilke verktøy som skulle brukes? (ev. samfunnspåvirkning)
 - Hvilke verktøy ble brukt?
 - Hvilke verktøy brukte du? (ev. standarder)
 - Hvilken rolle har de håndfaste (MS Office, Visio osv.) verktøyene spilt for det endelige produktet?
 - Essensielle for det endelige produkt?
 - Hvor viktige i forhold til RUP?
- 18. Hvor viktig er modellbyggingen i utviklingen føler du?**
- Viktigheten av modeller for det endelige produkt?
 - Ble det utviklet andre modeller enn de som er dokumenterte?
 - Hva med mengden modeller? Passe/for mange?
 - Var Use-case modelleringen et viktig verktøy i prosjektet? (for å se på om Cockburns kritikk av use-case modellering stemmer her)
- 19. Hvordan foregikk analysefasen?**
- Føler du at analysefasen gjorde at dere fikk en presis og detaljert oversikt over kravene?
 - På hvilken måte?

- 20. Hvordan vil du beskrive skillet mellom de ulike fasene?**
- Et klart skille?
 - Kommunikasjon mellom fasene bra?
- 21. Kvalitet: Var du involvert i testingen?**
Hvordan foregikk testingen? (står lite om dette i dokumentasjon fra BU)
- Som beskrevet i planen?
 - Spesielle teknikker?
 - Problemer med planen kontra praksis?
 - Fornøyd med testingen? Mulige forbedringer?
 - Finnes det systematisk gjennomlesing av dokumenter som en form for kvalitetssikring? (står nemlig ingenting om dette i vår dokumentasjon)
 - Annen form for kvalitetssikring som blir gjennomført?
- 22. Hva er din generelle oppfatning av arbeidsflyten/fasene i prosjektet?**
- Bedre kvalitet for produktet?
 - Effektivitet. Forskjell utover prosjektet?
 - Produktivitet
 - Hvordan opplever du skillet mellom teori og praksis?
- 23. Hvordan har tiden mellom iterasjonene blitt brukt? (Mellom Milepælene)**
- Ev. andre prosjekter?
 - Aktivitet hele tiden for å klargjøre ny iterasjon?
 - Hva skjer når en MP er avsluttet?
 - Stemning i gruppen. Har vi valgt riktig metodologi?
 - Hvordan er lengden på iterasjonene?
 - Hva er etter din mening den ideelle periode for en iterasjon?
- 24. Hvordan forholder du/dere til milepælene som er skissert i deres dokumenter?**
- Spiller de en stor rolle?
 - Hvilken form tar de? (reviews, publications, declarations)
 - Produkt vs. milepæler. Mellomproduktene (til hver iterasjon/milepæl) viktig for det endelige produkt?
- 25. Hvordan er det å jobbe i team i et slikt prosjekt?**
- Hvem har du samarbeidet med?
 - Hvordan har teamarbeidet fungert etter din mening?

- Hvordan har teaminndelingen foregått? (sturt av RUP, eller egen inndeling?)
- Team størrelse? Samsvar mellom dokumentasjon og virkelighet?
- Hvordan oppleves teamstørrelse?
- Hvordan spres info fra den ene til den andre?

26. Kommunikasjon:

Hvordan vil du karakterisere kommunikasjonen i prosjektet generelt?

- I forhold til annet arbeid (ikke RUP-arbeid) i organisasjonen?
- Kommunikasjonen sturt av RUP(kunstig) eller egne rutiner?
- Heftet/forbedret arbeidet?
- Møtefrekvens og form knyttet til prosjektet?
- Hvordan er tonen mellom medlemmene i prosjektgruppen?

27. Hvordan håndteres endringer i utviklingsprosessen?

- Hvordan håndteres endringer i utviklingsprosessen?
- Hva er ditt syn på RUP sin endringshåndtering?
- Hvem har kommet med endringsforslagene?
- Blir alle i bedriften hørt i samme grad? (hierarki ved endring)

28. Ved eventuelle oppsigelser eller utskiftinger i prosjektgruppen. Hvor mye arbeid tror du det hadde vært å sette inn en ny ansatt i prosjektet?

- Hvor mye hadde eventuelt blitt berørt?
- Tiltak for å få opp kompetansenivået hos ny ansatt?

29. Læring

- Hvordan vil du karakterisere arbeidet med RUP så langt når det gjelder egen utvikling?
- Hva har du lært?
- Bedre utbytte av måten du arbeider på nå enn tidligere metoder?
- Om du kunne endre arbeidssituasjonen din, hva ville du endret?
- Andre ting du vil trekke frem?

30. Ved neste prosjekt: Fortsette med RUP?

- Heller annen metodologi? Hvorfor?
- Tilpasse metodologi etter prosjekt?

31. Hvordan er følelsen i prosjektgruppen/din egen for det endelige resultatet og hittil arbeid?