

Systemutviklingsrutiner

”Motivasjon og forutsetning for videreføring av systemutviklingsrutiner”
En casestudie av en lokal avdeling i et større konsultentselskap

Kine Geliassen

Veileder: Even Åby Larsen

Masteroppgaven er gjennomført som ledd i utdanningen ved Universitetet i Agder og er godkjent som del av denne utdanningen. Denne godkjenningen innebærer ikke at universitetet inntår for de metoder som er anvendt og de konklusjoner som er trukket.

Universitetet i Agder, 2013

Fakultet for økonomi og samfunnsvitenskap

Institutt for informasjonssystemer

Forord

Denne rapporten er et resultat av forskning gjort i forbindelse med faget IS-501 på Universitetet i Agder våren 2013, og det er den avsluttende oppgaven gjennomført i forbindelse med Mastergrad i Informasjonssystemer. I dette studiet har jeg tilegnet meg kunnskap om tema som IT og ledelse, evaluering av IT-investeringer, e-Demokrati og eForvaltning samt Systemutviklingsprosesser og metoder. Det var dette sistnevnte emnet som utløste min interesse for forskningsområdet rutiner for systemutvikling, og dette ble dermed fokusområdet for meg.

Jeg har i løpet av semesteret forsket på hvordan rutiner i forbindelse med systemutvikling gjøres og hvordan man i en bedrift planlegger og gjennomfører disse rutinene. Kan man se at de lærer noe av de rutinene de bruker og hva slags motivasjon og forutsetninger har de for å velge å beholde de rutinene de har? Dette er det jeg har forsøkt å avdekke og forklare, ved å se nærmere på en lokal avdeling i et større konsulentselskap.

Det har vært svært verdifullt å se hvordan en slik avdeling bruker systemutviklingsrutiner i praksis og det har vært spennende å se forskjellene mellom hva man lærer i studiene og hvordan arbeidshverdagen faktisk er.

Avdelingen jeg har undersøkt har vært en uvurderlig kilde til informasjon og alle respondentene tok i mot meg med et smil og en velvilje som jeg ikke hadde turt å håpe på. Jeg vil med dette få rette en stor takk til bedriften og avdelingen som tok meg inn og lot meg stille spørsmål og observere de ansatte. Spesielt takk til avdelingslederen som møtte meg tidlig for å planlegge datainnsamlingen og som sendte møteinnkalling for intervju til alle respondentene.

Det som har vært gjennomgående vanskelig med skriveingen av rapporten har vært motivasjonen. Det har vært en spennende utfordring å skrive masteroppgaven individuelt og kanskje spesielt etter at jeg flyttet fra Kristiansand og dermed bort fra medstudenter. Mange ganger har prosessen vært frustrerende, men det er likevel tilfredsstillende å vite at jeg har gjort oppgaven alene og at jeg har klart det til slutt.

På tross av at jeg har flyttet bort fra medstudenter og veileder har jeg fått god hjelp gjennom forskningen, både personlig og over e-post og Skype. Jeg vil med dette få rette en stor takk til Førstelektor Even Åby Larsen for støtte og hjelp gjennom oppgaven. Han fortsatte å tro på meg også når jeg ikke trodde på meg selv.

Til slutt vil jeg også få takke menneskene rundt meg i hverdagen. Jan, min samboer for arbeidsro og oppmuntring på vanskelige dager og resten av min familie og venner for positive ord og støtte. I tillegg takk til Arnstein for korrekturlesing av oppgaven.

Sandnes, 6. juni, 2013

Kine Geliassen

Sammendrag

Bruken av metode i systemutvikling har lenge vært sett på som et viktig element for å kunne gjennomføre prosjekter raskt, strukturert, effektivt og forutsigbart. Noe som derimot har kommet frem rundt dette, er at nesten ingen bruker metoden helt slik den er tenkt og man ender opp med å gjøre tilpassninger. Videre forskning på hvorfor metoder fungerer eller ikke, er dermed problematisk ettersom alle bruker metodene forskjellig. Det jeg derfor har sett nærmere på, er i stedet rutiner og praksis.

Studien har tatt for seg to rutiner i en lokal avdeling i et større konsulentselskap, med fokus på testrutine og rutine for smidig utvikling. Det er blitt gjort kvalitativ forskning med enkelt case studie som utgangspunkt. Jeg gjorde en litteraturstudie i forkant av datainnsamlingen som la grunnlag for min forståelse av emnet og mitt ønske for videre forskning. Gjennom forskningen gjorde jeg også noe observasjon, men hovedfokus for innsamling av data var gjennom semistrukturerte intervjuer. Under intervjuene forsøkte jeg å avdekke så mye detaljer som mulig rundt de to rutinene i avdelingen jeg studerte.

Resultatene har blitt vurdert opp mot rammeverket ”A framework for describing ISD practice in context” i tillegg til problemstillingen. Problemstillingen gjennomgikk en viss utvikling basert på dataene jeg samlet inn og i startfasen ønsket jeg å kartlegge forskjeller mellom definerte og utførte rutiner og hvorvidt dette førte til bevisst endring og gjerne innføring av nye eller reviderte rutiner. Da resultatene viste at avdelingen ikke har gjennomført endringer i nyere tid ble det derfor heller interessant å se på: *Hvilken motivasjon og forutsetning har man for å videreføre systemutviklingsrutiner i en bedrift?*

Motivasjonen for videreføring av rutiner fremstod hovedsakelig som bekvemmelighet. Da både i form av at det er enkelt å slippe å endre noe, men også i form av at det ”alltid har vært sånn”. Motivasjon og forutsetninger henger nøye sammen og den viktigste forutsetningen for videreføring av rutiner er det faktum at de gjør nå, tydeligvis fungerer. At de velger å la være å gjennomføre endringer tyder på at de forventede konsekvensene av endring ikke veier opp for kostnadene og ressursene som er nødvendig for dette.

Mitt råd til avdelingen vil være å bruke mer tid på bevisstgjøring og evaluering av rutiner og arbeidsmåter slik at man er klar over hvorfor man jobber som man gjør. Får man faktisk en ønsket effekt, eller er man rett og slett for ”lat” til å finne ut om man burde drive rutineendring eller ei?

Videre ser jeg at generaliserbarheten av studien sannsynligvis er noe begrenset, men det er likevel interessant å se hvordan testrutinen som fremkom av intervjuene kanskje også kan tas i bruk av andre bedrifter da den fremstår som relativt enkel og ressursbesparende. Päiväranta et al., (2011) beskriver ønsket om en rutinedatabase og testrutinen jeg har avdekket i min forskning kan ses på som et potensielt bidrag til denne. Studien bidrar også med en mer generell bevisstgjøring rundt temaet og gir en dypere forståelse av rutiner og spesielt rutiner sett opp mot metode og hvordan man kan lære av seg selv og erfaringer.

Innholdsfortegnelse

| | | |
|-------|--|----|
| 1 | Innledning..... | 1 |
| 1.1 | Bakgrunn | 1 |
| 1.2 | Problemstilling..... | 2 |
| 1.3 | Motivasjon | 2 |
| 1.4 | Avgrensing..... | 3 |
| 1.5 | Strukturen i rapporten | 3 |
| 2 | Teori og litteraturgjennomgang..... | 4 |
| 2.1 | Innsamling av litteratur..... | 4 |
| 2.2 | Metode | 5 |
| 2.2.1 | Agile metoder - smidig utvikling | 6 |
| 2.2.2 | Scrum | 7 |
| 2.2.3 | Tilpassning av metode; Methods-in-Action | 9 |
| 2.3 | Framework for building theory from ISD practices | 9 |
| 2.3.1 | Rutine | 10 |
| 2.4 | NIPO-grid | 11 |
| 2.5 | Ametodisk..... | 11 |
| 2.6 | Rutineendring | 12 |
| 2.7 | Kunnskapsarbeid | 12 |
| 2.8 | Prosessforbedring - SPI | 12 |
| 2.9 | Organisasjonsl ring og taus kunnskap..... | 13 |
| 2.10 | Testrutiner | 14 |
| 3 | Forskningstiln ring..... | 15 |
| 3.1 | Argumentasjon og filosofiske antagelser | 15 |
| 3.2 | Forskningsdesign | 16 |
| 3.2.1 | Valg av case | 16 |
| 3.2.2 | Respondenter fra casebedriften | 17 |
| 3.2.3 | Intervju | 17 |
| 3.2.4 | Observasjon | 18 |
| 3.2.5 | Transkribering av data..... | 18 |
| 3.2.6 | Analyse av data | 19 |
| 3.3 | Validering av funn | 20 |
| 3.4 | Begrensninger i min forskningsdesign | 21 |
| 3.5 | Etiske hensyn og anonymisering | 22 |
| 4 | Forskningskontekst..... | 23 |
| 5 | Resultater..... | 26 |

| | | |
|-------|---|----|
| 5.1 | Testrutine | 26 |
| 5.1.1 | Bruk av testplan..... | 26 |
| 5.1.2 | Når det testes første gang | 27 |
| 5.1.3 | Ulike typer test | 27 |
| 5.1.4 | Loggføring av test og resultater..... | 28 |
| 5.1.5 | Når kunden involveres i test..... | 28 |
| 5.1.6 | Likhet i testrutine fra prosjekt til prosjekt | 29 |
| 5.1.7 | Kodegjennomgang - andre tester din kode..... | 29 |
| 5.1.8 | Kriterier for godkjenning og kategorier av feil | 30 |
| 5.2 | Rutine for smidig systemutvikling | 30 |
| 5.2.1 | Arbeidsfordeling..... | 31 |
| 5.2.2 | Møtevirksomhet | 31 |
| 5.2.3 | Fremtidig prosjektrum..... | 32 |
| 5.2.4 | Endringsønsker og uforutsette hendelser | 33 |
| 5.2.5 | Leveranser | 33 |
| 5.2.6 | Evaluerings/kvalitetssikring..... | 33 |
| 5.2.7 | Hva gjør bedriften smidig?..... | 34 |
| 5.2.8 | Hvem legger føringene? | 34 |
| 5.3 | Analyse av data..... | 35 |
| 5.3.1 | Testrutine..... | 35 |
| 5.3.2 | Rutine for smidig systemutvikling | 37 |
| 5.3.3 | Kontekstuelle elementer som påvirker rutinene | 40 |
| 6 | Diskusjon..... | 42 |
| 6.1 | Funn sett opp mot rammeverket | 42 |
| 6.2 | Funn sett opp mot problemstillingen | 47 |
| 6.3 | Bidrag til praksis, råd og anbefalinger til avdelingen..... | 48 |
| 6.4 | Svakheter og begrensninger i datamaterialet | 49 |
| 6.5 | Teoretiske implikasjoner | 50 |
| 7 | Konklusjon og implikasjoner | 51 |
| 8 | Referanser..... | 52 |
| 9 | Vedlegg | 54 |
| 9.1 | Vedlegg 1 - Intervjuguide..... | 54 |

Figurliste

| | |
|---|----|
| Figur 1 - A framework for describing ISD practice in context (Päivärinta et al., 2011)..... | 1 |
| Figur 2 - Scrum, iterative sykluser (Schwaber & Microsoft, 2004, p. 6)..... | 8 |
| Figur 3 - Method-in-Action (Fitzgerald et al., 2002, p. 12) | 9 |
| Figur 4 - Utnitt og eksempel på resultattabell | 20 |
| Figur 5 - A framework for describing ISD practice in context (Päivärinta et al., 2011)..... | 42 |

Tabelliste

| | |
|-------------------------------|----|
| Tabell 1 - Datakilder | 23 |
| Tabell 2 - Respondenter | 24 |

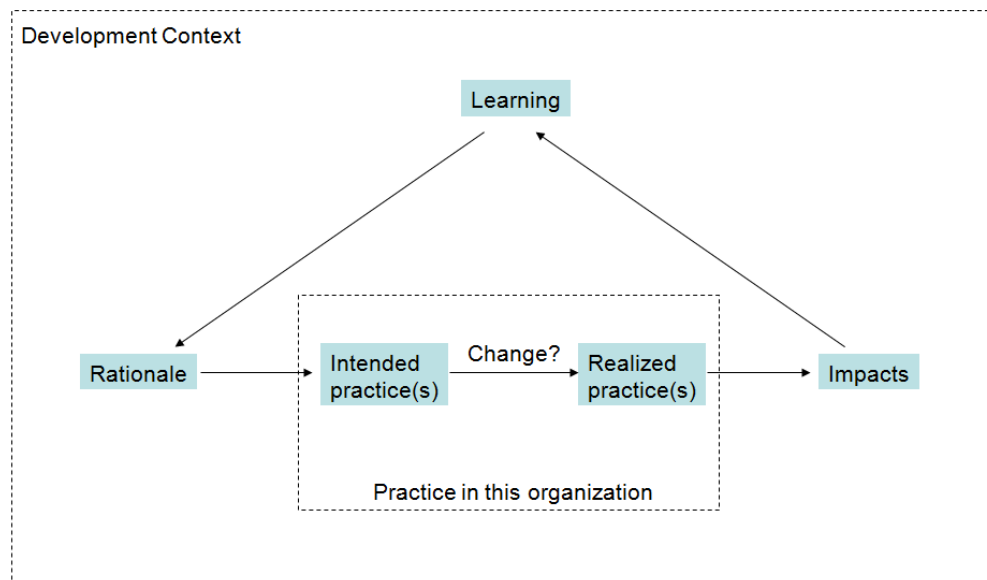
1 Innledning

I dette kapitlet vil jeg introdusere tema for oppgaven og gi en bakgrunn for hvorfor forskningen er interessant. Jeg vil videre kort presentere rammeverket som oppgaven i stor grad bygger på samt problemstillingen jeg har valgt. Videre kommer jeg til å si noe om min motivasjon for forskningen samt avgrensning i studien og til slutt vil jeg presentere hva som kommer i resten av rapporten.

1.1 Bakgrunn

Bruken av metode i systemutvikling har lenge vært sett på som et viktig element for å kunne gjennomføre prosjekter raskt, strukturert, effektivt og forutsigbart. Man har en rekke metoder, både smidige og plan-drevne og systemutviklere kan velge blant en rekke alternativer. En interessant observasjon som derimot har kommet frem rundt dette, er at nesten ingen bruker metoden helt slik den er tenkt. Metoder vil sjelden passe perfekt til flere ulike prosjekter og man ender dermed opp med tilpassede metoder som Fitzgerald et. al., (2002) definerer som "methods-in-action". Dette vil si at man gjerne tar utgangspunkt i en formalisert metode, men at denne endres for å fungere i den gitte situasjonen.

Videre forskning på hvorfor metoder fungerer eller ikke fungerer, er dermed problematisk ettersom alle bruker metodene forskjellig. Dette forsøker Päivärinta et al., (2011) å gripe fatt i ved å gå bort fra å studere metoden, til i stedet å se på rutiner og praksis. Ved å se på rutiner ser man heller på hvordan bedriftene faktisk gjennomfører oppgaver og ikke hva metoden sier at de skal gjøre. Definerte rutiner og utførte rutiner er ikke alltid det samme og Päivärinta et al., (2011) presenterer en modell for hvordan man kan lære av å se på systemutviklingspraksis:



Figur 1 - A framework for describing ISD practice in context (Päivärinta et al., 2011)

På bakgrunn av denne modellen ser man at det er en rekke elementer som påvirker praksisen i en organisasjon. Man har en begrunnelse for å gjøre noe på en viss måte, man har en definert rutine og en utført rutine. Her kan det ha skjedd en endring og de utførte rutinene får noen

effekter. Bedriften kan lære av disse og vil kanskje dermed endre begrunnelsen for å gjøre noe på en viss måte. Alt dette er påvirket av kontekst, både organisasjonskontekst og utviklingskontekst.

Det kan se ut som det er gjort mye forskning på området rundt endring og forbedring av systemutviklingsrutiner, endringsledelse og hvordan man skal effektivisere rutiner og prosesser. Men det fremgår et hull i forskningen på hva som gjør at noen velger å videreføre rutinene i stedet for å gjennomføre endring. Med min problemstilling vil jeg forsøke å bidra til mer forståelse på dette området.

1.2 Problemstilling

Den opprinnelige tanken for prosjektet og caset var en antagelse om at det ville finnes noen klart definerte rutiner fra organisasjonens side og at disse i utgangspunktet skulle følges. Videre antok jeg at rutinene jeg fikk beskrevet fra respondentene ville være delvis avvikende fra dette. Tanken var å undersøke forskjellen mellom de definerte rutinene og de utførte rutinene for å se om dette førte til en bevisst endring og da se på hva som motiverte denne forandringen eller innføringen av nye rutiner. Jeg ville avdekke hvilke forutsetninger bedriften ville ha for å innføre testrutiner eller andre rutiner. I forskningsforslaget var derfor problemstillingen som følger:

”Hvilken motivasjon og forutsetninger har man for innføring av nye testrutiner i en bedrift?”

Det som derimot viste seg når jeg startet datainnsamlingen var at ikke bare hadde bedriften lite definerte og detaljerte rutinebeskrivelser. Men respondentene hadde ikke opplevd endring i rutiner som de kunne huske eller beskrive. Det ble dermed heller interessant å se på hva som gjorde at de valgte å videreføre de rutinene hadde. Den reviderte problemstillingen ble derfor:

”Hvilken motivasjon og forutsetning har man for å videreføre systemutviklingsrutiner i en bedrift? Med fokus på testrutine samt rutine for smidig utvikling”

For å kunne svare på denne problemstillingen er det viktig at jeg definerer hva jeg legger i begrepene ”motivasjon” og ”forutsetning” og jeg vil derfor gi noen korte definisjoner på dette her. Rutinebegrepet blir nærmere undersøkt og beskrevet i litteraturgjennomgangen.

Motivasjon: Det som forårsaker handling og aktivitet for mennesker. Faktorene som styrer mål og mening og som gir oss engasjement.

Forutsetning: Noe som er påkrevd for at noe annet skal skje. Hva er betingelsen for at en gitt hendelse skal finne sted.

1.3 Motivasjon

Motivasjonen min for dette temaet stammer i hovedsak fra interessen for emnet IS-402 Systemutviklingsprosesser og metoder og IS-411 Aktuelle tema i systemutvikling. Jeg har gjennom tiden min på universitetet lært mye om ulike metoder og hvordan disse kan brukes for å drive med systemutvikling på en bedre måte. Likevel har jeg også sett at når man faktisk implementerer disse metodene, enten i studentprosjekter eller i bedrifter, så ender man opp med å måtte tilpasse dem til egen situasjon. Da virker det meningsløst at ulike metoder studeres og vurderes som gode eller dårlige når man gjør det på bakgrunn av bruk som ikke

inngår i metodebeskrivelsen. Kanskje man heller burde studere virkeligheten? Det som faktisk gjøres i systemutviklingsprosjekter?

Inntrykket mitt er at fokus i faktiske systemutviklingsprosjekter ikke ligger så mye på metoden og hvorvidt den brukes riktig eller ei, men heller på rutiner og ulike praksiser for å oppnå best mulig resultater med minst mulig ressurser. Dette er motivasjonen min for å se på rutinene i en bedrift, fremfor å se på en hel metode.

1.4 Avgrensing

En av de viktigste avgrensningene jeg har gjort i oppgaven er at jeg har valgt å gjøre en studie av et enkelt case fremfor å velge flere. Dette ble gjort i stor grad fordi ønsket om å gå i dybden er sentralt, i tillegg til risikoen for at et flercase design kan bli for overfladisk med tanke på tiden jeg har hatt til rådighet. Jeg innser at jeg ideelt sett burde sett på flere case for å kunne gi et fullstendig svar på forskningsspørsmålet, men med tiden jeg hadde tilgjengelig ønsket jeg heller å gå i dybden i ett case fremfor å pirke i overflaten på flere.

En annen klar avgrensning jeg har gjort er valget om å fokusere på to spesifikke rutiner fremfor å ta for meg hele metoden. Dette har blitt gjort fordi jeg ved å ta for meg to rutiner kan i mye større grad gå inn i dybden på disse rutinene. Jeg kan forsøke å forstå akkurat hvordan de arbeider, hvorfor de gjør det på denne måten og hva som motiverer de ansatte og ledelsen til å fortsette å gjøre det slik.

1.5 Strukturen i rapporten

I rapporten presenteres det syv hovedkapitler med en rekke underkapitler og jeg vil her kort presentere innholdet i disse.

Kapittel 1: Innledning. Her presenterer jeg bakgrunnen for oppgaven og valget av tema, jeg viser til rammeverket som resultatene vil vurderes opp mot samt en presentasjon av problemstillingen, motivasjon samt avgrensninger gjort i oppgaven.

Kapittel 2: Teori og litteraturgjennomgang. Dette er en relativt omfattende litteraturgjennomgang hvor jeg presenterer forskningen som allerede er gjort på området samt at det setter bakgrunnen for resten av oppgaven og gir en forståelse for emnet. Jeg har også her definert de viktigste begrepene som går igjen i rapporten.

Kapittel 3: Forskningstilnærming. Her presenterer jeg hvordan jeg har gått frem for å samle inn data og hva slags forskningsmetode jeg har valgt. I tillegg beskrives validering, begrensninger i forskningsdesignet samt etiske hensyn.

Kapittel 4: Forskningskontekst. I dette kapitlet presenterer jeg caset og avdelingen jeg har undersøkt. Jeg sier litt om respondentene og hvordan data ble samlet inn.

Kapittel 5: Resultater. Resultatene er delt inn i tre hoveddeler. Testrutine og dens underkategorier, rutine for smidig utvikling og dens underkategorier samt analyse av resultatene.

Kapittel 6: Diskusjon. Her har jeg satt resultatene opp mot litteraturen og da spesielt rammeverket. I tillegg har jeg sett på funn i forhold til problemstilling. Praktiske og teoretiske implikasjoner beskrives også, i tillegg til begrensninger i datamaterialet.

Kapittel 7: Konklusjon. Til slutt oppsummerer jeg de viktigste funnene og svarer på problemstillingen.

2 Teori og litteraturgjennomgang

I dette kapittelet ønsker jeg å legge grunnlaget for forståelsen av studien og hva jeg fant da jeg snakket med respondentene. Mange elementer rundt hvordan systemutvikling gjøres og rutine og metodene man bruker, kan fremstå som forvirrende da man har en rekke ulike begrep som brukes om hverandre og som i ulike situasjoner kan ha ulike meninger. Jeg vil videre derfor forsøke å komme med noen begrepsforklaringer, samt gi et generelt overblikk over litteraturen på emnet systemutviklingsrutiner og hvordan dette er blitt beskrevet og forstått tidligere.

Jeg vil videre gi definisjoner på begrep som metode, ametodisk, rutine, smidig samt prosessforbedring og flere andre for å gi en forståelse av jungelen av begrep som brukes når det er snakk om hvordan man driver systemutvikling. Jeg vil i tillegg forsøke å gi en dypere forståelse av det som beskrives som et rammeverk for å forstå praksis og rutiner i systemutvikling (Päivärinta et al., 2011) da dette er i stor grad hva rapporten bygger på.

2.1 Innsamling av litteratur

Litteraturen jeg presenterer er blant annet samlet inn gjennom ulike kurs jeg har deltatt i ved Universitetet i Agder, da med spesielt fokus på IS-402, IS-411 og andre utviklingskurs med fokus på metoder. I tillegg har jeg gjort litteratursøk på Internett og disse ble gjort i hovedsak med tjenestene Google Scholar og Scopus. Universitetets nettverk ble benyttet for å få fullverdig tilgang til disse verktøyene.

For å finne passende artikler på Internett brukte jeg en rekke søkeord og mange ulike kombinasjoner av disse i mine søk. Dette er ordene jeg brukte:

- ISD
- Method
- Knowledge work
- Routines
- Organizational routines
- Organizational learning
- Amethodical
- Change, continuous change
- Scrum
- Agile development, agile manifesto
- Test plan
- Test routines
- Communication
- Organization

I tillegg til søk på Internett med bruk av disse ordene ble det også gjort en rekke søk etter spesifikke artikler og forskere, som dukket opp i andre nyttige artiklers litteraturlister. Tips til litteratur fra veileder var også en nyttig kilde for informasjon samt bruk av pensumlister fra de to spesielt relevante fagene IS-402, og IS-411. Henholdsvis bøkene (Cockburn, 2007),

(Fitzgerald et al., 2002) og (Boehm & Turner, 2003) fra IS-402 samt en mer omfattende artikkelsamling fra IS-411.

Litteratur ikke direkte knyttet til rapportens emne ble også i stor grad funnet på Internett og basert på tips fra veileder og andre forelesere. Dette var litteratur som omhandlet forskningsmetode, hvordan gjennomføre intervjuer på best mulig måte, analysemetoder og lignende.

Litteraturen jeg har samlet inn og presentert er i stor grad gjort for å gi en bakgrunn for å forstå forskningen og hensikten med denne, i tillegg til et ønske om å kunne se mine funn opp mot tidligere litteratur. Da med spesielt fokus på rammeverket i Figur 1 som jeg vil sammenligne i enda større grad med mine funn i bedriften. Litteraturen ble i tillegg brukt til dels som et utgangspunkt for intervjuguiden. Ikke nødvendigvis for spesifikke spørsmål, men som bakgrunn for min egen forståelse av hva som kunne være nyttig og interessant.

Det kan se ut som det er gjort mye forskning på området rundt endring og forbedring av systemutviklingsrutiner, endringsledelse og hvordan man skal effektivisere rutiner og prosesser. Men det fremgår et hull i forskningen på hva som gjør at noen velger å beholde og videreføre rutinene. Litteraturen jeg derfor presenterer er mer beskrivende av de perifere påvirkene, men jeg vil forsøke å bruke denne litteraturen for å belyse hva som kan være motivasjonen og forutsetningene for å *ikke* endre på rutinene.

Lyytinen og Robey (1999) beskriver organisasjoners manglende evne og vilje til å lære av erfaringer på en noe drastisk måte. Man kan spørre seg om bedrifter som ikke ønsker å endre seg har gått i denne fellen eller om de er berettiget fordi de har funnet noe som fungerer.

“Organizations fail to learn from their experience in systems development because of limits of organizational intelligence, disincentives for learning, organizational designs and educational barriers. Not only have many organizations failed to learn, but they have also learned to fail. Over time they accept and expect poor performance while creating organizational myths that perpetuate short-term optimization” (Lyytinen & Robey, 1999, p. 85).

2.2 Metode

Som nevnt tidligere er det gjort en mengde forskning på ulike metoder og hvorvidt de fungerer godt eller dårlig. Problemet er at ytterst få evner eller ønsker å bruke metodene fullt og helt på den måten de er tenkt. Bruken av metoder blir for eksempel beskrevet som at: *“Adherence to methods was regarded by default as useful; methods were to be rigorously pre-defined and utilized as intended by the method developers.”* (Päivärinta, Smolander, & Larsen, 2012 p. 2). På bakgrunn av dette ser man at fokuset på metoder som løsningen, kanskje kan ha vært forhastet. Jeg mener derfor at det kan det være nyttig å se bort i fra de teoretiske arbeidsmetodene og heller fokusere på det som faktisk gjøres; nemlig praksisen. For å gi en bakgrunn for denne tanken vil jeg begynne med å presentere prinsippene rundt metode.

En metode for å drive systemutvikling kan defineres på mange måter, en av definisjonene man kan bruke er denne:

“A coherent and systematic approach, based on a particular philosophy of systems development, which will guide developers on what steps to take, how these steps should

be performed and why these steps are important in the development of an information system” (Fitzgerald et al., 2002, p. 5).

Dette er en definisjon som tydelig beskriver hvordan en metode dikterer hvordan man skal jobbe på en systematisk måte med visse tankesett som skal hjelpe utviklere å vite stegene videre i utviklingen. En alternativ definisjon viser hvordan man kan ”koke det ned” til at: *“A methodology is the conventions that your group agrees to”* (Cockburn, 2007, p. 149). I min forskning har jeg tenkt på metode som en sammenstilling av disse. På den ene siden er metode et enkelt begrep som sier noe om regelverket deltagerne forholder seg til. På den andre siden beskrives mange metoder som svært rigide, hvor hensikten er å ha klare svar på hva som er det neste steget.

I en metode har man en rekke faste elementer som brukes i teamarbeid. De kan fortolkes ulikt fra de forskjellige metodene, men alle vil forsøke å dekke over disse strukturelle elementene på en eller annen måte. Cockburn (2007) beskriver 13 elementer. Jeg vil kort nevne og forklare disse her:

- **Roles:** rollen man har i teamet i forhold til hvem som gjør hva, hvilke evner og personlighetstrekk de har
- **Skills:** evnene man trenger for å gjennomføre rollen
- **Teams:** rollene som jobber sammen under ulike omstendigheter
- **Techniques:** prosedyrene folk bruker for å gjennomføre oppgaver
- **Activities:** hvordan folk bruker dagene sine
- **Process:** hvordan aktiviteter passer sammen, ofte med pre- og postbetingelser for aktivitetene
- **Work Products:** hva man produserer
- **Milestones:** hendelser som markerer fremgang eller fullføring
- **Standards:** reglene teamet adopterer for verktøy, produkter og beslutningstaking
- **Quality:** kvaliteten på både aktiviteter og produkter
- **Team Values:** resten av metodeelementene som styres av teamets verdier
- I tillegg har man elementene **Personality** og **Tools** som også påvirker (Cockburn, 2007, pp. 150-152).

2.2.1 Agile metoder - smidig utvikling

Det finnes mange ulike metodetyper og man skiller gjerne mellom såkalte plandrevne og smidige metoder. De plandrevne metodene beskrives gjerne som;

“...characterized by a systematic engineering approach to software that carefully adheres to specific processes in moving software through a series of representations from requirements to finished code” (Boehm & Turner, 2003, p. 10).

Dette regnes gjerne som den mer tradisjonelle måten å utvikle på og man bruker ofte mer dokumentasjon og klare krav og spesifikasjoner for produktet og hvordan man jobber. Fossefallsmetoden nevnes ofte i forbindelse med plandrevne metoder. Her går man gjennom klare steg som ikke gjentas; analyse for å etablere en kravspesifikasjon, design, koding, implementering og testing (Boehm & Turner, 2003).

Agilt, eller på norsk; smidig, er en annen retning innenfor systemutvikling som går på at man skal bort fra det tidligere konseptet med tunge fossefallsmetoder og heller over på å bedre håndtering av endring. Det smidige prinsippet kan beskrives slik:

”Agile methods encourage programmers to shed their heavyweight process chains, embrace change, and escape into agility. Advocated methods have short cycle times, close customer involvement, and an adaptive rather than predictive mind set” (Boehm & Turner, 2003, p. 4).

Konseptet smidig utvikling kommer fra det såkalte ”Agile Manifesto”, laget i 2001. De viktigste elementene av smidighet beskrives i manifestet på denne måten:

***”Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan***

That is, while there is value in the items on the right, we value the items on the left more.” (Beck et al., 2001)

Dette viser til hvordan man ønsker et større fokus på individene, muligheten for kreativitet og hvordan man arbeider, ansvar for eget arbeid, tettere samarbeid med kunden slik at man forstår hverandre bedre, samt en villighet og evne til å håndtere endringer. På denne måten ønsker man å gi bedre og mer velfungerende programvare.

Man har en rekke ulike typer smidige utvikling og jeg kan nevne metodeeksempler som Crystal Clear, Extreme Programming (XP), Kanban og Scrum som noen av de mer kjente. Jeg vil gjøre en videre definisjon av Scrum da dette er av spesiell interesse for oppgaven.

2.2.2 Scrum

Scrum er en av metodene som er spesielt interessant i denne casestudien fordi dette er i utgangspunktet metoden som ligger til grunn for de rutinene virksomheten bruker. Dette er også metoden som ble brukt i utstrakt grad av avdelingen før oppkjøpet. Der i mye større grad enn hva metoden brukes nå. Jeg vil derfor definere noen av de viktigste elementene slik at begrepene respondentene bruker er klare.

Som vist i Figur 2 kan man se at hele det generelle prinsippet bak Scrum er bruken av iterasjoner og at man i starten av denne ser gjennom hva man må gjøre fra Product Backlog, teamet velger det som lar seg gjennomføre i løpet av iterasjonen og så gjør man det. Man møtes hver 24. time for å snakke om status og så lar man teamet arbeide isolert med prosjektet frem til en del av funksjonaliteten er klar på slutten av iterasjonen. Den kreative prosessen hvor medlemmene i teamet får tenke selv er en viktig del av Scrum (Schwaber & Microsoft, 2004).



Figur 2 - Scrum, iterative sykluser (Schwaber & Microsoft, 2004, p. 6)

I Scrum har man noen artefakter eller dokumenter som produseres og brukes i arbeidsprosessen. Som tidligere nevnt har man **Product Backlog** som er en prioritert liste med oppgaver, disse har estimert tidsbruk og tilsvarer alle delene av produktet man ønsker å produsere. Dette dokumentet utvikler seg ettersom man ser at forutsetningene eller målene endrer seg. I tillegg har man **Burndown Graph** som viser hvor mye arbeid som gjenstår sett i forhold til hvor mye tid man har i løpet av en sprint. En **Sprint** kan defineres som et gitt tidsrom, på vanligvis 30 dager, hvor man utvikler en bestemt del av funksjonaliteten. I forbindelse med sprint bruker man også gjerne en **Sprint Backlog** som har en liste med oppgaver fra Product Backlog som skal produseres i den gitte sprinten (Schwaber & Microsoft, 2004).

I tillegg har man en rekke møter som brukes i Scrum. I forbindelse med sprint har man **Sprint Planning Meeting** som starter en sprint hvor man bestemmer hvor mye av Product Backlog man tror man kan klare å gjennomføre og plasserer dette i Sprint Backlog. **Sprint Retrospective** avholdes på slutten av en sprint og er for å diskutere hva som gikk bra og hva som gikk dårlig i den foregående sprinten. Man ønsker å avdekke hva man kan gjøre bedre i neste sprint. **Sprint Review** er et møte hvor man demonstrerer det som er blitt produsert i sprinten for kunden. Kun ferdig funksjonalitet vises frem. Møtene som avholdes hver 24. time kalles gjerne for **Daily Scrum Meeting**. Dette er som nevnt et statusmøte hvor man diskuterer hva man holder på med synkroniserer arbeid (Schwaber & Microsoft, 2004).

Et annet viktig konsept i Scrum er de ulike rollene man bruker for å definere ansvarsområder. Man har **Team** som er utviklerne og de som jobber med å utvikle software i hver sprint. **Product Owner** er ofte kunden og er personen som har ansvar for å få mest mulig verdi ut av Product Backlog. Denne personen representerer alle Stakeholders i prosjektet. En **Stakeholder** er en person som har interesse i prosjektet, enten fordi de betaler, fordi de skal bruke produktet eller fordi de blir påvirket på annet vis. **Scrum Master** er en viktig rolle som har ansvar for at scrum-prosessen gjennomføres riktig og på en måte som maksimerer fordelene ved metoden samt at han/hun skal holde distraksjoner og byråkrati unna teamet slik at de skal kunne fokusere på utviklingen (Schwaber & Microsoft, 2004).

2.2.3 Tilpassning av metode; Methods-in-Action

Litt av utfordringen og det som er interessant med bruk av en valgt metode er at man veldig ofte ender opp med å gjøre tilpassninger for at den skal kunne brukes i en gitt situasjon og under gitte omstendigheter.

“In actual development practice, formalized ISD methods are rarely applied in their entirety, nor as originally intended by their creators, although they may provide a template to guide development practice.” (Fitzgerald et al., 2002, p. 13)

Med bakgrunn i denne forståelsen for hvordan man bruker metode så har Fitzgerald et al., (2002) satt opp et rammeverk som forklarer det forfatterne kaller Method-in-Action. Som man ser på Figur 3 så ser man at det er en rekke forhold som spiller inn på hvordan man bruker en metode. Man starter gjerne med en formalisert metode som da kan bli påvirket av en rekke elementer som for eksempel: utviklingskonteksten, rollen metoden har og spiller i prosjektet i tillegg til at utviklerne i seg selv ofte vil ha en form for innflytelse.

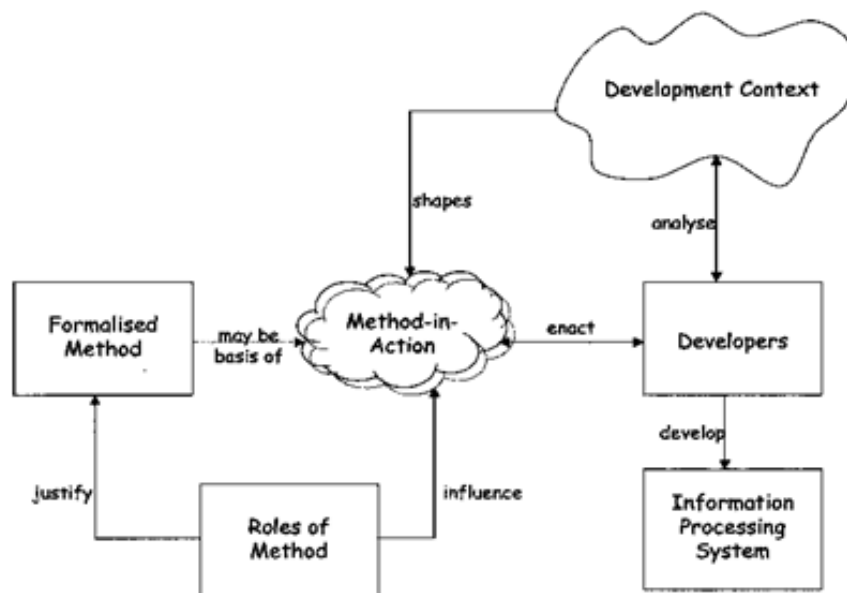


Fig. 2.1 A Framework for ISD Method Use

Figur 3 - Method-in-Action (Fitzgerald et al., 2002, p. 12)

På bakgrunn av denne modellen og konseptet method-in-action kan man se behovet for en ny type forskning med et nytt fokus. Det er ikke lenger like interessant å forske på bruk av metode når de færreste utnytter metoden slik som skaperen har tenkt. Dette fører meg over på rammeverket som går bort i fra fokuset på metode og heller over på rutine og hvordan man kan lære av og forbedre rutinene for systemutvikling.

2.3 Framework for building theory from ISD practices

Rutiner fra systemutvikling oppstår gjerne fra tidligere erfaring, det vil si effekter man ser i andre prosjekter eller fra teori. Målet er å forbedre rutinene man bruker samt å gjøre systemutviklingen mer effektiv og generelt sett bedre. Bakgrunnen for rammeverket er blant annet Fitzgerald et al., (2002) og andre forfatteres skiftende fokus bort fra metode og over på rutiner og prosesser.

Rammeverket fra Figur 1 beskrives som en måte for å lage teori om systemutvikling. Man bruker i modellen seks begreper som legger grunnlaget for hvordan man best kan forstå og utnytte rammeverket for å lage teori og forhåpentligvis forbedre systemutviklingen. Man snakker om;

- **Læring (learning):** Man må kunne se på utviklingsprosjekter som eksperimenter hvor man forsøker å teste den teorien man bruker for utvikling, slik at man bedre kan forstå hvorfor noe fungerer eller ikke fungerer.
- **Rutine (a practice):** Noe man gjør ofte eller daglig. Man kan skille mellom organisasjonsrutine, som er hvordan en organisasjon bruker kunnskapen tilgjengelig, og vanlige rutiner som er hvordan man gjør en oppgave. En rutine trenger ikke være detaljert definert på samme måte som en metode.
- **Utviklingskontekst (development context):** Konteksten utviklingen skjer i. Man snakker da både om selve utviklingen, samt organisasjons- og miljøkontekst. Struktur, kultur, kunder, teknologi etc. vil alle påvirke utviklingskonteksten.
- **Begrunnelse (rationale):** Rettferdigjørelsen man bruker for de rutiner som brukes i organisasjonen.
- **Effekt (impact):** Alle effekter og bi-effekter man får av å bruke rutinen uavhengig om de blir målt eller ikke. (Eller i observert/oppdaget i det hele tatt.)
- **Teori (theory):** Teorier som kan analysere, beskrive og forklare rutiner i systemutvikling (Päivärinta et al., 2011).

"The framework locates ISD practices in a learning loop that is situated in a development context." (Päivärinta et al., 2011, p. 611). Det er dette rammeverket som legger utgangspunktet for min forskning og som ble presentert i introduksjonen som Figur 1. Jeg vil i min forskning ha et spesielt fokus på det som figuren presenterer som såkalte "intended practices", "realized practices". I tillegg vil hvilke effekter man får av rutine og hvorfor man i dette caset velger å beholde sine rutiner fremfor å ha en bevisst bruk av læringsløkka være viktig.

2.3.1 Rutine

Rutine er et begrep som er spesielt interessant for denne rapporten og jeg ønsker derfor å beskrive dette i noe mer detalj. Forskningen bygger på rammeverket av Päivärinta et al., (2011) og det er derfor naturlig for meg å bruke denne artikkelens forståelse av begrepet. På engelsk bruker man ordet "practice" og jeg har valgt å oversette dette til rutine da begrepet praksis kan virke forvirrende. Dette fordi jeg har sett på et fast arbeidsområde som test og bruken av smidighet og ikke enkeltstående elementer som man gjerne i større grad kan tolke ut fra begrepet praksis. Artikkelen beskriver hvordan man i en utviklingsammenheng gjerne får en sammenheng mellom utviklingsrutine og organisatoriske rutiner.

"In context of a development project or an organization, a systems development practice may become an organizational practice or routine, which can be defined as the organization's routine use of knowledge, especially 'know-how'" (Päivärinta et al., 2011, p. 613).

Videre får man også en forklaring på hvordan man kan skille mellom hva som kan defineres som en metode og hva man kan kalle en rutine.

“If we compare a systems development method and a practice, a method adopted in an organization always embodies a predefined practice or a set of them, whereas a practice is not always defined at the detailed level, at least with regard to all potential elements of method knowledge” (Päivärinta et al., 2011, p. 613)

En rutine beskrives som noe man gjør rutinemessig, det vil si på ca samme måte hver gang. Den kan være basert på en formell beskrivelse, for eksempel i en metode, men ikke nødvendigvis.

2.4 NIPO-grid

Et annet eksempel på rammeverk som studerer hvordan rutiner gjennomføres og hvordan de endres i forhold til hvem som bruker og hvem som planlegger, kan være ”The NIPO Grid” hvor man ser på rutiner og i hvor stor grad de brukes (Larsen, Päivärinta, & Smolander, 2008). Dette kan beskrives på mange måter som rammeverket som la grunnlaget for utviklingen av rammeverket i kap 2.2 og man kan se visse likhetstrekk, men også et naturlig tillegg av elementer som forklarer læringen og årsaker for endring.

NIPO-grid er først og fremst et visualiseringsredskap for utbredelse og endringer av utbredelsen til rutiner. Det forsøker å se på praksis og endring i systemutvikling og på hvordan den blir påvirket av hva som planlegges og hva som faktisk gjøres i tillegg til at det sier noe om på hvilket nivå i organisasjonen denne praksisen defineres. Rammeverket skiller praksis i to dimensjoner; beregnet (intended) og faktisk brukt (actually used), videre kan praksisen ha beregnet bruk eller faktisk bruk i fire nivåer: ikke i det hele tatt (not at all) **N**, individuelt **I**, prosjekt-nivå **P** og hele organisasjonen **O** (Larsen et al., 2008).

Videre er tanken at man setter de to typene praksis opp mot hver av de fire nivåene slik at man ser hvilke deler av organisasjonen som planlegger en gitt praksis og om de samme delene eller andre deler faktisk bruker praksisen i det daglige arbeidet. På denne måten får man en bevisstgjøring og en synlighet av hvordan systemutvikling fungerer i praksis i en gitt organisasjon. Man vil dermed lettere kunne se mønstre for å avdekke hvordan man kan forbedre seg.

2.5 Ametodisk

En annen retning man kan velge, som motsetning til bruk av metode eller en tilpasset metode er det man kaller ametodisk tilnærming. Truex et al. (2000) har gjort forskning på tekster om metode hvor man forsøker å påpeke et annet alternativ til tilpassede metoder; nemlig *ametodisk* tilnærming. Dette beskrives som et negativt konstrukt av hva som er essensielt ikke metodisk. Det vil si at ametodisk betyr styring og gjennomføring av programmering uten en predefinert sekvens, kontroll, rasjonalitet eller påstand om at det er universelt.

“An amethodical development activity is so unique and unpredictable for each information systems requirement that even the criteria of contingent development methods are irrelevant.” (Truex et al., 2000, p. 54)

Forfatterne setter de klassiske antagelsene om metodisk programmering, forstått ut fra forskningstekster, opp mot det som dermed kan forstås som ametodisk, nemlig det som ikke blir sagt. De elementene de da definerer som ametodiske er som følger: ISD er tilfeldig, opportunistisk og prosessdrevet av uhell. De skriver at ISD er samtidige prosesser som er

overlappende i tillegg til at det er mellomrom. Det foregår i helt unike og ideografiske former samt at det er forhandlet, kompromissbasert og lunefullt (Truex et al., 2000).

2.6 Rutineendring

Endring av rutiner er et tema som er forsket mye på, men da ofte med spesielt fokus på *organisasjonsrutiner* som kan defineres som; *"patterned sequences of learned behavior involving multiple actors who are linked by relations of communication and/or authority"* (Cohen & Bacdayan, 1994, p. 555). Selv om jeg skal se på systemutvikling spesielt så kan jeg ikke anta at dette er så særegent at litteratur om organisasjonsrutiner ikke er gjeldene for mye av det jeg skal se på. Rutiner i organisasjon ses i litteraturen også i mange tilfeller på som en kilde til endring, fleksibilitet og forbedring (Feldman, 2000; Feldman & Pentland, 2003).

Endring av rutiner er i første omgang erfaringsbasert og skjer i forbindelse med forandringer i erfaring. Det beskrives av Espedal (2006, p. 468) med *"...how multiple actors in an organization generate and change routines in the face of changing experience"*. Dette støtter opp om hvordan modellen til Päivärinta et. al., (2011) er satt opp og hvordan man gjennom en læringsløkke endrer rutiner basert på erfaring og endring, lærer noe nytt og forsøker kontinuerlig å forbedre rutinene.

2.7 Kunnskapsarbeid

Kunnskapsarbeid er også et begrep som dukker mye opp i litteraturen om systemutvikling og det kan i følge Iivari et. al., (2001) defineres slik;

"... we propose four criteria to distinguish KW from other work: (i) KW is based on a BoK, (ii) entails working on representations (data) of the objects of work, (iii) stipulates a deep, theoretical understanding of the objects of work, and (iv) KW produces results, which entail knowledge as their essential ingredient." (p. 1027).

BoK står her for Body of Knowledge, som betyr et komplett sett av begreper, termer og aktiviteter som utgjør et profesjonelt domene. Med dette kan man forstå at kunnskapsarbeid er arbeid hvor kunnskap er den viktigste verdien arbeiderne har. Årsaken til at dette er spesielt interessant i forbindelse med systemutvikling, er fordi man som programmerer har mulighet til kreativitet i utførelsen av arbeid. Rutiner som favner kreativitet og kunnskap som viktigste verdi kan være vanskeligere å beskrive og standardisere enn rutiner for masseproduksjon av en fysisk vare for eksempel.

2.8 Prosessforbedring - SPI

Prosessforbedring kan være et interessant område å se på i forhold til rammeverket fordi det kan inngå som en del av læringsdelen. Man ønsker å lære av rutinene og prosessene man bruker, og om mulig forbedre disse. Dette kan gjøres med enkle forandringer i rutiner eller som større, virksomhetsomspennende prosessforbedringer.

Prosessforbedring innenfor systemutvikling, eller såkalt Software Process Improvement (SPI) blir ofte beskrevet som komplekse prosjekter i stor skala hvor man setter i gang virksomhetsomspennende endringsinitiativer (Ngwenyama & Nørbjerg, 2010). Med denne

beskrivelsen får man et inntrykk av hvor komplisert et slikt prosjekt kan være, og man har mange utfordringer å overkomme for å gjennomføre et vellykket endringsprosjekt.

Herbsleb et al., (1997) nevner en rekke elementer som går igjen i SPI-prosjekter som ikke var vellykket, altså utfordrende områder. Listen som beskrives ser slik ut:

- *“High levels of “organizational politics”.*
- *Turf guarding.*
- *Cynicism from previous unsuccessful improvement experiences.*
- *Belief that SPI “gets in the way of real work”.*
- *Need more guidance on how to improve, not just what to improve.”* (p. 39)

Motstand og motvilje fra de ansatte kan også være en stor utfordring i et SPI-prosjekt (Ngwenyama & Nørbjerg, 2010; Pries-Heje, Aaen, & Elisberg, 2008). Når SPI settes i gang og man planlegger endringer som skal gjennomføres i organisasjonen og i prosessene, ligger mye fortsatt opp til de ansatte. Dersom de ansatte velger ikke å ta i bruk endringene som foreslås vil man sannsynligvis ende opp med prosessforbedring som ikke blir gjennomført.

SPI er et problematisk område med mengder av utfordringer. Et av hovedproblemene med prosessforbedring er at man er avhengig av mange elementer samt andre mennesker for at det skal fungere på en god måte. God kommunikasjon er essensielt og uten samarbeid med hele bedriften er det vanskelig å oppnå gode resultater med SPI.

2.9 Organisasjonslæring og taus kunnskap

Læring er en viktig del av rammeverket og det kan derfor være interessant å se på begrepet ”organizational learning” som jeg har valgt å oversette til *organisasjonslæring*.

Businessdictionary.com (2013) definerer begrepet slik:

“Organization-wide continuous process that enhances its collective ability to accept, make sense of, and respond to internal and external change. Organizational learning is more than the sum of the information held by employees. It requires systematic integration and collective interpretation of new knowledge that leads to collective action and involves risk taking as experimentation” (businessdictionary.com, 2013).

Essensen i begrepet er altså forståelsen av hvordan organisasjoner lærer og tilpasser seg i gitte situasjoner. Det handler ikke bare om kunnskapen de ansatte innehar men også om organisasjonen som helhet og dens evne til å lære av seg selv og hendelser i hverdagen. I organisasjonslæring er det viktig at man er selvbevisst og at man ønsker å forbedre seg. *“Continuous improvement is still important to a learning organization, because it is the foundation of a learning organization, and the basis of organizational capacity”* (Wang & Ahmed, 2003, p. 14).

Videre har man forskere som beskriver hvordan såkalt taus kunnskap kan være viktig for organisasjonslæring:

“Much recent attention has focused, in particular, on the importance of 'tacit knowledge' for sustaining firms' competitiveness, and its role in technological innovation and organizational learning.” (Lam, 1998, p. 1)

Taus kunnskap kan defineres som “... *knowledge that cannot be easily articulated or transferred because it is uncodified and context specific*” (Lam, 1998, p. 1). Dette viser til hvordan man gjerne som ansatt innehar kunnskap som ikke kan klart defineres og dokumenteres fordi den er usagt. Det handler mer om hvordan man som menneske oppfatter gitte situasjoner og at man med erfaring lærer hva som er passende eller ikke.

2.10 Testrutiner

Jeg vil her forsøke å gi en kort oversikt over de ulike begrepene som brukes av respondentene rundt testrutiner og de ulike testene som brukes.

Testplan kan defineres som et dokument eller en plan som skal beskrive tilnærmingen man har til test, prosedyren for gjennomføring med tidsramme, kriterier for godkjenning samt testens vilkår og begrensninger (Turbit, 2006).

Unit-test, eller såkalt **enhetstest** kan beskrives som en type test som ofte skrives av utviklerne selv for å sikre at små enheter av kode fungerer som ønsket. Man tester små komponenter og det gjøres ofte med såkalte test-script som utvikleren kan kjøre gjennom (Page, Johnston, & Rollison, 2009).

Integrasjonstest er ganske enkelt en test som kjøres for å sikre at den nye koden man har skrevet fungerer sammen med den resterende koden eller de andre systemene den skal integreres med (Page et al., 2009).

Akseptansetest ut mot kunden kan beskrives som en test hvor målet er å bekrefte at systemet følger spesifikasjonen og at systemet oppfører seg som avtalt. Kunden får teste systemet selv og bestemmer om han/hun synes det er tilfredsstillende (Hansen & Hjertø, 2003).

3 Forskningstilnærming

I dette kapittelet vil jeg presentere min forskningstilnærming. Jeg har valgt å bruke kvalitativ forskning hvor jeg har gjort en casestudie i en lokal avdeling av et større konsultentselskap. Data har blitt samlet inn ved bruk av semistrukturerte intervjuer og til dels observasjon som verktøy.

“The importance of social issues related to computer based information systems has been recognized increasingly over the last decade, and this has led some IS researchers to adopt empirical approaches which focus particularly on human interpretations and meanings.” (Walsham, 1995, p. 74)

Med bruk av kvalitativ forskning får man et iboende fokus på de mer myke dataene som blir påvirket av de menneskelige faktorene og hvordan man kan tolke og forstå meninger. Det blir en mer fortolkende forskning som gir både respondentene og forskerne rom for egne oppfatninger av fenomenet.

“Whereas quantitative data deals with numbers, qualitative data deals with meanings. Meanings are mediated mainly through language and action.” (Dey, 1993, p. 11)

På bakgrunn av min problemstilling fremstod det som naturlig å velge kvalitativt fremfor kvantitativt for å kunne gå i dybden i ett enkelt case for å kartlegge erfaringene som de ansatte gjorde seg rundt systemutviklingsrutiner.

Videre i kapittelet vil jeg vil først si noe om min argumentasjon for valg, mine filosofiske antagelser og deretter presentere forskningsdesignet med fokus på intervju, observasjon, transkripsjon og hvordan jeg har gjort analysen. Mot slutten av kapittelet vil jeg også ta frem valg av case, hvordan jeg gjorde validering, potensielle begrensninger i min forskningsdesign samt etiske problemstillinger og anonymisering.

3.1 Argumentasjon og filosofiske antagelser

Å gjøre forskning kan kort defineres som å skape noe ny kunnskap som er tilfredsstillende for brukerne av forskningen (Oates, 2006). Videre beskrives det man kan definere som *god* akademisk forskning;

“... we do not jump to conclusions, but carefully find sufficient and appropriate sources of data, properly record, analyze and interpret that data, draw well-founded conclusions based on the evidence, and present the findings in an acceptable way in a report, thesis, conference presentation or journal article.” (Oates, 2006, p. 5)

Målet mitt har vært å gjøre så *god* forskning som mulig ved å bruke gjennomtenkte datakilder, samle disse inn på en best mulig måte, analysere og tolke data godt, samt å forsøke å trekke fornuftige konklusjoner. For å besvare problemstillingen i min forskning, valgte jeg å bruke case studie som tilnærming. Ved å bruke case studie fikk jeg muligheten til å undersøke bedriften mer i dybden og finne ut hva slags rutiner de brukte, samt at jeg fikk gjort dybdeundersøkelser rundt hvordan to spesifikke rutiner ble gjort. Rutinene jeg valgte for nærmere undersøkelse var testrutine og rutine for smidig systemutvikling.

Med tanke på de filosofiske antagelsene i min forskning fremtrer positivismen som begrep. Her har man to nøkkelantagelser, den ene er at verden er ordnet og ikke tilfeldig og den andre er at jeg som forsker kan studere den objektivt. Positivismen er den klassiske måten å forske på og tanken er at man oppnår kunnskap gjennom vitenskapelig observasjon av "virkeligheten". Man antar dermed at det finnes en virkelighet man kan forstå. Tilnærmingen jeg har til forskningen inkluderer også en deduktiv tankegang. Deduktiv tilnærming vil si at man går ut fra bakenforliggende teorier om tema på forhånd som man kan skaffe empiri om og deretter ser om kunnskapen man får, stemmer overens med teoriene og hypotesene (Oates, 2006).

For å oppsummere min holdning til de filosofiske antagelsene kan jeg si at jeg hadde noen formening og noe kunnskap på forhånd som jeg ønsket å bekrefte, men jeg hadde i tillegg et ønske og en åpenhet for å finne nye elementer som ikke fremtro tydelig i litteraturen.

3.2 Forskningsdesign

En case studie kan defineres på denne måten:

"A case study is an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident." (Yin, 2009, p. 18)

Oates (2006) beskriver at hensikten med en case studie er å; *"obtain a rich, detailed insight into the 'life' of that case and its complex relationships and processes"* (Oates, 2006, p. 141). Mitt mål med forskningen var å gå i dybden hos én enkelt organisasjon og undersøke i en naturlig setting der jeg benyttet meg av flere kilder og metoder for min datainnsamling. Dette er alle trekk som kjennetegner en case studie (Oates, 2006).

Min case studie er en blanding av deskriptiv og forklarende. Det vil si at jeg i analysen av data har forsøkt å både forklare motivasjon og forutsetninger, men at jeg også har brukt tid på å beskrive rutine og hvordan de forholder seg (Oates, 2006). Jeg har valgt et enkeltcase design fordi ønsket om å gå i dybden er sentralt, i tillegg risikerer jeg at et flercase design kan bli for overfladisk med tanke på tiden jeg har til rådighet. I tillegg får jeg med bruken av ett enkelt case et tydeligere og mer sammenhengende bilde på hvorfor rutine er som de er, og hvilke situasjoner og organisasjonelle faktorer som har ført til at de ikke endres.

Ønsket om fordypning i ett enkelt case er også årsaken til valg av to enkle rutiner fremfor en mer helhetlig oversikt over hvordan avdelingen arbeider. Ved å ta for meg to rutiner kan jeg i mye større grad gå inn i dybden på disse rutine og forsøke å forstå akkurat hvordan de arbeider, hvorfor de gjør det på denne måten og hva som motiverer de ansatte og ledelsen til å fortsette å gjøre det slik.

3.2.1 Valg av case

Valg av bedrift for min casestudie ble gjort på bakgrunn av en rekke forhåndsbestemte kriterier og noen uforutsette faktorer. Jeg ønsket først og fremst en bedrift som drev med systemutvikling. Dette var sentralt for meg ettersom målet var å studere rutine som ble brukt og planlagt i forbindelse med utvikling.

Videre ønsket jeg en bedrift av en viss størrelse slik at jeg ville ha muligheten til å snakke med et tilstrekkelig antall respondenter som jobbet med de samme rutine, samt at størrelsen

på bedriften også ville påvirke i forhold til organisasjon og ledelse. Da i den forstand at man i små bedrifter ofte har en flat struktur uten særlig overhead. Jeg ønsket å se på en avdeling under et større konsern hvor det var en mulighet at jeg ville finne ulikheter mellom definerte (fra ledelsen) og utførte rutiner. I tillegg antok jeg større sannsynlighet for pålegg om mer spesifikke rutiner i en større bedrift, enn i en liten.

I tillegg til de teoretiske kravene hadde jeg også noen mer praktiske krav til case. Dette var at det helst måtte være en bedrift som jeg kunne reise til fysisk for å møte respondentene da jeg ikke ønsket å være avhengig av telefonmøter eller møter over Internett så langt det var mulig.

3.2.2 Respondenter fra casebedriften

Respondentene i undersøkelsen har vært 10 ansatte fra en lokal avdeling til et større konsultantselskap. Jeg har snakket med seks utviklere og fire personer fra ledelsen. Av utviklerne var det to som jobbet med såkalt frontend-utvikling og design samt to i fra det de kaller backend. (Frontend beskrives som utvikling som har med design og brukergrensesnitt å gjøre. Altså det man enkelt kan se på skjermen uten at man må kjøre gjennom en funksjon. Backend ble beskrevet mer som det som skjer bak kulissene i systemet. Altså det som brukeren ikke vil se.) Videre snakket jeg med to andre utviklere som også i stor grad jobbet med backend, men som var 100 % utleid som konsulenter til et prosjekt i et annet stort firma. Disse beskrev for det meste hvordan rutiner gjennomføres der de er utleid, men det ble også tidvis diskutert hva de er vant med fra sitt eget firma. I tillegg til utviklere snakket jeg også med ansatte fra ledelsen. Her snakket jeg med to avdelingsledere hvorav den ene også fungerte som prosjektleder for øyeblikket, samt en ren prosjektleder som i tillegg hadde bakgrunn som scrum-master. Til slutt snakket jeg også med øverste leder for avdelingen. Se kapittel 4 for videre beskrivelse av bedriften og forskningskonteksten.

3.2.3 Intervju

Min primære kilde for datainnsamling har som nevnt vært semistrukturerte intervjuer.

“The qualitative interview is used in qualitative research of all kinds, whether positivist, interpretive or critical. It is used in case studies, in action research, in grounded theory studies, and in ethnographies.” (Myers & Newman, 2007, p. 2)

Et kvalitativt intervju beskrives av som spesielt godt egnet i situasjoner ”når relativt få enheter undersøkes, når vi er interessert i hva det enkelte individ sier og når vi er interessert i hvordan den enkelte fortolker og legger mening i et spesielt fenomen” (Jacobsen, 2005, p. 130). I mitt tilfelle har caset vært begrenset til 10 enheter, eller respondenter og målet var å avdekke hva den enkelte mente om to rutiner.

Jeg gjennomførte 10 intervjuer på mellom 30 og 60 minutter på kontoret til bedriften i løpet av uke 10. Min kontaktperson hadde sendt møteinnkallinger til alle de ansatte som skulle intervjues og jeg fikk dermed gjennomført samtlige intervjuer på relativt kort tid. Intervjuene ble gjort med seks utviklere og fire personer fra ledelsen. Av utviklerne var det to som var utleid på oppdrag hos kunde og disse møtte opp på kontoret etter arbeidstid for å bli intervjuet. Ett av intervjuene ble avholdt ved bruk av Skype. Dette var fordi denne utvikleren satt på en av konsultantselskapets andre avdelinger, men var likevel ansatt i, og jobbet for, avdelingen som jeg studerte. Ved bruk av intervjuer som datainnsamling er det viktig at man

også tenker på det menneskelige aspektet av at ikke alle håndterer å bli intervjuet på samme måte. (Kvale, Brinkmann, & Anderssen, 2009)

Det ble skrevet en intervjuguide for å hjelpe meg å holde oversikt under intervjuene og denne ble skrevet med hensikten å avdekke så mye detaljer som mulig rundt rutinene. Jeg stilte i tillegg noen spørsmål om respondenten og prosjektet de jobbet med for å gi kontekst.

Jeg valgte å bruke ganske spesifikke spørsmål i intervjuguiden (som man kan se i Vedlegg 1), men ved å bruke semistrukturerte intervjuer var det viktig for meg at respondentene også kunne komme med kommentarer og tanker om tema som jeg ikke hadde tenkt på. Jeg hadde derfor en liste med temaer jeg ville innom, samt de spesifikke spørsmålene, men startet ofte intervjuene med å bare be de om å fortelle meg litt om hvordan de gjorde rutinene. Med arbeidsrutiner som tema var det viktig for meg å kunne være lydhør for intervjuobjektene med tanke på at de kjente bedriften og rutinen mye bedre enn jeg gjorde.

Alle intervjuene ble innledet med en forklaring om hvem jeg var, hva jeg ønsket å oppnå med intervjuet samt hva det skulle brukes til. I tillegg ble intervjuobjektene informert om etiske retningslinjer og at de ville være anonyme. Jeg ba om tillatelse til å gjøre lydopptak og startet dette. Jeg gjorde to separate lydopptak av hvert intervju for å forsikre at jeg fikk med meg alt. Jeg tok også noe notater underveis i intervjuene, både for å huske detaljene bedre, men også for å gi meg selv noe å se på av og til for å ikke virke konfronterende.

3.2.4 Observasjon

I forbindelse med gjennomføring av intervjuene på avdelingen fikk jeg muligheten til å tilbringe tre arbeidsdager i bedriften. I løpet av disse dagene satt jeg i kontorlandskapet og kunne overfladisk observere hvordan de ansatte kommuniserte i tillegg til at jeg fikk tillatelse til å delta på et statusmøte.

Å observere kan kort defineres som 'å se på' eller 'å være oppmerksom på' noe man ønsker å studere. (Oates, 2006) Målet mitt var å se og forstå hvordan de ansatte på avdelingen jobbet. Muligheten jeg fikk til å observere statusmøte ga meg en fin introduksjon til avdelingen og mange av mine intervjuobjekter da dette møtet var det aller første jeg gjorde da jeg ankom bedriften.

“In participant observation, the researcher takes part in the situation under study, so that it can be experienced from the point of view of the others in that setting.” (Oates, 2006, p. 208)

Jeg utførte en variant av deltagende observasjon og de ansatte var klar over hvem jeg var og at de ble observert. Jeg ble introdusert rundt bordet og fikk forklart av avdelingsleder som ledet møtet, hva som skjedde og hva jeg så på skjermen. Jeg deltok ikke aktivt i diskusjonen, men hadde mulighet til å stille spørsmål underveis. Jeg tok observasjonsnotater både under møtet og i løpet av dagene jeg satt på kontoret.

3.2.5 Transkribering av data

Etter gjennomføringen av intervjuene og observasjonen på avdelingen brukte jeg nesten tre uker på å transkribere alle lydopptakene jeg hadde gjort. Å transkribere én time med opptak ville i følge Oates (2006), normalt ta fem timer, eventuelt lenger dersom man er uerfaren eller har dårlig kvalitet på opptakene. Jeg beregnet derfor seks timer per lydopptak for å være på

den sikre siden. Dette viste seg å være et greit estimat og jeg brukte ca én arbeidsdag per intervju. Noen intervjuer var mer utfordrende enn andre å transkribere da respondentene hadde ulik dialekt samt at noen snakket svært raskt og andre uklart, nølende eller upresist.

”Transkripsjon fra lydopptak til tekst er forbundet med en rekke tekniske og fortolkningsmessige problemstillinger - spesielt angående ordrett talespråkstil versus skriftspråkstil - som det ikke finnes mange standardregler for men snarere en rekke valg som skal treffes.” (Kvale et al., 2009, p. 189)

Valgene jeg tok under transkribering var å skrive det i enkelt format uten bruk av spesialtegn og lignende for å indikere tonefall og lyder. Jeg unnlot å transkribere bruken av ord som; eh, ah, hmm og lignende lyder for å få mer flyt i teksten og språket. Jeg forsøkte likevel å forholde meg til lydopptakene så ordrett som mulig, samtidig som jeg ønsket å holde setningene logiske og forståelige også som skriftlig språk. På denne måten har jeg kunnet beholde særegenheten til de ulike respondentene i sitatene, i tillegg til at språket også fungerer skriftlig.

3.2.6 Analyse av data

Det finnes en rekke alternative måter å gjøre analyse på. Med kvalitativ forskning hvor man gjør intervjuer får man ofte inn en stor mengde data som etter transkripsjon utgjør mange sider tekst som man i ettertid skal klare å trekke kunnskap ut fra.

“There is no one kind of qualitative data analysis, but rather a variety of approaches, related to the different perspectives and purposes of researchers. To distinguish and assess these different perspectives fully would be a formidable and perhaps rather fruitless task, particularly as the boundaries between different approaches and their relation to what researchers actually do when analyzing data is far from clear.” (Dey, 1993, p. 1)

Etter transkribering av mine 10 intervjuer samt noen notater fra observasjon satt jeg igjen med ca.80 sider tekst. For å kunne trekke noe fornuftig informasjon ut av dette brukte jeg mye tid på gjennomlesning, markering og notering i transkripsjonen. For å gi meg selv de beste praktiske forutsetningene skrev jeg ut hele transkripsjonen på papir med litt ekstra brede marger slik at jeg hadde plass for å skrive ned tanker og stikkord mens jeg analyserte.

I første omgang leste jeg gjennom dokumentet for å få et overblikk over informasjonen jeg hadde og for å mentalt kunne dele den inn i tre segmenter. Det vil si at én del av stoffet var relativt uinteressant for min forskning, noe var mer generell informasjon om bedriften og det siste segmentet var det mest relevante for mitt forskningsspørsmål. På bakgrunn av denne inndelingen kunne jeg lettere jobbe videre med den delen som var mest interessant for min forskning (Oates, 2006).

Etter å ha identifisert den delen av informasjonen som jeg ønsket å jobbe videre med, brukte jeg temaanalyse for å finne de viktigste temaene og kategoriene som ble tatt opp gjennom kontakten med bedriften. Temaene ble satt inn i en tabell hvor jeg plasserte alle disse i ulike kolonner og respondentene i rader. Se Figur 4 for eksempel på hvordan denne tabellen ble satt opp. Temaene ble funnet induktivt ved at jeg var åpen for hva enn som dukket opp i teksten. Deretter gikk jeg over transkripsjonen en andre gang hvor jeg i tabellen kort kunne identifisere alle respondentenes meninger og utsagn om de ulike temaene. Etter en slik

oppsummering var det enklere å identifisere klare enigheter og uenigheter blant respondentene samt å legger merke til de overordnede tendensene i teksten.

| Respondent | Testrutine | | | | | | | Rutine for smidig utvikling | | | | | | | | |
|------------|------------------|----------------------------|------------------|----------------------------------|------------------------------|---------------------|---|---|------------------|----------------|------------------------|--|------------|-----------------------------|----------------------------|------------------------|
| | Bruk av testplan | Når det testes første gang | Ulike typer test | Loggføring av test og resultater | Når kunden involveres i test | Likhet i testrutine | Kodegjennomgang - Andre tester din kode | Kriterier for godkjenning og kategorier av feil | Arbeidsfordeling | Møtevirkosmhet | Fremtidig prosjekttrom | Endringsønskter og uforutsette hendelser | Leveranser | Evaluering/ kvalitetsikring | Hva gjør bedriften smidig? | Hvem legger føringene? |
| #1 | | | | | | | | | | | | | | | | |
| #2 | | | | | | | | | | | | | | | | |

Figur 4 - Utnitt og eksempel på resultattabell

Under gjennomlesning av transkripsjonen brukte jeg også markeringstusjer for å tydeliggjøre interessante funn og jeg brukte ulike farger blant annet for å identifisere potensielt gode sitater samt andre farger for markering av andre interessante utsagn og spennende tema.

3.3 Validering av funn

Å validere betyr at man forsøker å sikre troverdigheten i forskningen man gjør.

“If a piece of research has validity, it means that an appropriate process has been used, the findings do indeed come from the data and they do answer the research question(s).” (Oates, 2006, p. 10)

Datainnsamlingen i min forskning har først og fremst basert seg på intervjuer. Ved bruk av intervjuer som primær datakilde risikerer man å få såkalt dårlig reliabilitet fordi *“the effect of the researcher and the context mean that consistency and objectivity are hard to achieve.”* (Oates, 2006, p. 198) For å bedre reliabiliteten i min forskning har jeg forsøkt å ha stor gjennomsiktighet ved å dokumentere min fremgangsmåte og funn i slik detalj at forskningen kan gjentas med de samme resultatene. I tillegg har jeg prøvd å være bevisst på min egen rolle som forsker i den forstand at jeg har forsøkt å stille spørsmål uten å lede respondentene samt at jeg har oppfordret til å komme med egne tanker og meninger. I analysen har jeg etter beste evne forsøkt ikke å tolke for mye etter mine egne meninger og erfaringer, men har heller ønsket å finne hva respondentene egentlig sa.

For å sikre troverdighet av resultatene valgte jeg å bruke lydopptak fordi man alltid kan gå tilbake til opptaket for å sikre at man husker riktig informasjon. Lydopptak i tillegg til notater fra intervjuene bidro også sannsynligvis til reduksjon av feiltolkning og misforståelser av

innsamlet informasjon i ettertid. Andre grep jeg har brukt for å spesifikt bedre validiteten av både intervjuer og observasjoner i forskningen er:

- **Sitat:** Ved å bruke direkte sitater fra lydopptakene, slik at det ikke er min forståelse/hukommelse av hva som ble sagt som presenteres, men akkurat hva respondenten faktisk sa.
- **Triangulering:** At jeg intervjuet flere personer om de samme tingene. På denne måten kan jeg se om det jeg finner ut er felles for flere personer eller situasjoner, eller om de er helt særegne.
- **Refleksjon:** Jeg har forsøkt å være bevisst på meg selv og min rolle i forskningen. Det vil si at jeg har reflektert over hvordan jeg eventuelt påvirker intervjuobjektene eller situasjonen med mine egne tanker, meninger og erfaringer (Oates, 2006)
- **Member validation:** Dette vil si at man bruker intervjuobjektene selv for å forsikre seg om at man har forstått informasjonen riktig. Under intervjuene brukte jeg små oppfølgings spørsmål underveis for å sikre at jeg hadde forstått respondenten riktig (Bygstad & Munkvold, 2007). Denne typen validering ble brukt både under innsamling av data og i analysedelen da jeg tidvis sendte e-post til respondentene for å sikre at min forståelse stemte.

3.4 Begrensninger i min forskningsdesign

Det finnes visse iboende begrensninger i valget av case studie som forskningstilnærming. Ettersom jeg kun har studert én bedrift og kun har hatt 10 respondenter så kan dette ha ført til et noe snevert bilde av situasjonen. Kunnskapen jeg har fått vil ikke nødvendigvis være generaliserbar ettersom det er akkurat denne bedriftens rutiner, bakgrunnen for disse og hvordan de gjøres som er blitt forsket på. Likevel kan det tenkes at andre bedrifter bruker lignende rutiner og at noe felles kunnskap kan trekkes ut.

Andre begrensninger ved kvalitativ forskning er faren for påvirkning fra forskeren, altså mine meninger og fortolkninger. Det har vært viktig for meg å gjøre mitt beste for ikke å la mine egne meninger sette for mye farge på forskningsresultatet. Likevel er påvirkning fra forskeren en del av hvordan kvalitativ forskning nødvendigvis må bli. Man kan ikke helt objektivt kategorisere informasjon innhentet gjennom intervjuer og jeg har bakgrunn av dette hatt en rekke utfordringer i forbindelse med dataanalysen.

Med tanke på at min forskning i hovedsak er gjort basert på 10 intervjuer med ansatte i bedriften ser jeg også at alle mine resultater er avhengig av at respondentene har snakket sant og så ærlig som mulig. Jeg har spurt om hva slags arbeidsrutiner de bruker i tillegg til hva slags rutiner de vet organisasjonen og ledelsen ønsker at de skal bruke. Med slike spørsmål risikerer jeg at noen av respondentene har ”pyntet litt på sannheten” for og unngå å fremstå som at de ikke følger reglene. Det er en vanskelig situasjon å unngå helt som forsker og jeg har som motgrep til dette blant annet spurt hva de spesifikt gjorde sist de testet for eksempel. Dette ble gjort for å tvinge respondenten til å huske akkurat hva de gjorde og fortelle om en spesifikk hendelse, noe som gir mindre mulighet til å dikte eller å fortelle hva de tror jeg vil høre.

Noen siste utfordringer er nærmere knyttet til kvalitativ forskning i seg selv. Eksempler på dette er at man gjennom kvalitative studier får inn store mengder informasjon som kan være

vanskelig å komme gjennom og utfordrende å finne tema og mønster i. Dataanalysen risikerer å være svært fortolkende og konklusjoner man trekker kan derfor bli mer foreløpige (Oates, 2006).

3.5 Ethiske hensyn og anonymisering

I en forskningssituasjon har deltakerne visse rettigheter som man som forsker må opprettholde. Dersom bedriften for eksempel ønsker at forskningen skal være anonym og/eller konfidensiell er det viktig at jeg respekterer dette og at jeg er forsiktig med spredningen og lagringen av informasjon jeg samler inn. Det vil si at jeg må avklare på forhånd i hvor stor grad informasjonen kan deles med resten av klassen, forelesere og andre, i tillegg til at jeg ikke må være uforsiktig med hvordan jeg oppbevarer dokumenter og lydopptak.

I utgangspunktet hadde jeg tillatelse til å bruke bedriftens navn og detaljer i rapporten, men jeg valgte heller å anonymisere caset. Dette ble gjort på bakgrunn av en rekke ting. Først og fremst fordi inkludering av bedriftsnavn i rapporten var forutsatt kontroll av dokumentet før publisering. Det er i utgangspunktet ikke problematisk, men jeg ønsket ikke å legge ytterligere press på meg selv ved å måtte gi bedriften tid til gjennomlesning og godkjenning for så å risikere at den *ikke* ble godkjent og at jeg dermed måtte brukt mye tid på anonymisering i ettertid. I tillegg til denne noe egoistiske årsaken for anonymisering ser jeg også at hvordan man utfører arbeid og arbeidsrutiner kan skape en tidvis uheldig representasjon av bedriften og enigheten blant respondentene. I tillegg anser jeg ikke bruk av bedriftens navn som relevant i den forstand at det kan påvirke casestudiets utfall eller resultater. Det er dermed ikke forstyrrende at jeg lar være å identifisere caset.

4 Forskningskontekst

Virksomheten ble valgt som case studie først og fremst fordi det var et større selskap som drev med systemutvikling. Det var viktig for meg å velge en bedrift som hadde et team av utviklere jeg kunne intervju i tillegg til at jeg måtte ha en viss tilgang til ledelsen for å kunne sammenligne de definerte rutineene med de utførte. Ønsket om å gjøre denne slags sammenligning førte også til at jeg ønsket en større virksomhet hvor det ville være rimelig å anta en viss overhead fra organisasjonens ledelse.

En annen årsak for valg av denne bedriften var at jeg håpet at dette skulle være en bedrift hvor de ansatte hadde krav og pålegg fra konsernledelsen om hvordan de ønsket at arbeidet skulle gjennomføres og hvilke rutiner som skulle brukes. At oppkjøpet nettopp var satt i verk var også en av årsakene til at jeg fant akkurat dette caset spesielt interessant.

Datakildene som ble brukt i caset detaljeres i Tabell 1:

Tabell 1 - Datakilder

| Datakilde | Type | Lengde | Metode |
|----------------------------------|-----------------------|------------------------------|--------------------------------|
| Frontend 1 | Intervju | 22 min | Personlig møte |
| Frontend 2 | Intervju | 22 min | Personlig møte |
| Backend 1 | Intervju | 52 min | Over Skype |
| Backend 2 | Intervju | 27 min | Personlig møte |
| Utleid/Back 1 | Intervju | 1 time og 5 min | Personlig møte |
| Utleid/Back 2 | Intervju | 1 time og 10 min | Personlig møte |
| Rådgiver | Intervju | 49 min | Personlig møte |
| Avdelingsleder | Intervju | 45 min | Personlig møte |
| Avdelingsleder/ Prosjektleder | Intervju | 54 min | Personlig møte |
| Leder | Intervju | 49 min | Personlig møte |
| Ukentlig statusmøte | Møteobservasjon | 1 time | Observasjon med notater |
| Kontorlandskap | Observasjon av kontor | 1 time, fordelt over 3 dager | Generelt overblikk med notater |

Min case studie ble gjort av en lokal avdeling i et av Norges største konsultentselskap. I denne avdelingen har jeg studert og snakket med 10 respondenter, hvorav fire var fra ledelsen og seks var utviklere. Akkurat denne avdelingen var tidligere tilhørende et annet firma som ble kjøpt opp av virksomheten i 2012. Alle respondentene bortsett fra én utvikler, hadde tidligere jobbet sammen i denne bedriften før den ble kjøpt opp og inkorporert i det nåværende selskapet. De fleste medlemmene på teamet hadde altså jobbet sammen mellom to og seks år

da de ble intervjuet. Dette viste seg i at de fleste ansatte ga uttrykk for at de kjente hverandre godt og at de visste hvordan de jobbet best og hvordan de jobbet best sammen.

Videre følger en tabell som er satt opp for å gi et kort overblikk over respondentene og deres stilling i firma. I tillegg til stilling beskriver jeg hva slags prosjekt de jobber på, hvor lenge de har vært ansatt i bedriften samt utdanning. Tabellen har også som hensikt å gi navn til de ulike respondentene slik at jeg enklere kan henvise til dem fremover. Respondentnavnene jeg har valgt er ikke nødvendigvis beskrivende for det eneste de jobber med, men det er det de i hovedsak jobber med i tidsperioden hvor de ble intervjuet. Det beskrives i tillegg fra Leder at dette teamet i denne tidsperioden jobber med ca 23-24 ulike kunder/prosjekter.

Tabell 2 - Respondenter

| # | Respondent | Stilling | Antall år ansatt | Prosjekt | Utdanning |
|----|-------------------------------|---|------------------|---|---|
| 1 | Frontend 1 | Seniorkonsulent | 3 år | Flere prosjekter, diverse sharepoint, samt intranett. | Almennlærer og bachelor i Informatikk |
| 2 | Frontend 2 | Webdesigner/ utvikler | 2 år | To prosjekter, nettsider | Dataingeniør |
| 3 | Backend 1 | Konsulent | 2 år | Tre prosjekter, portal, nettside, app, nettbutikk | Bachelor i Informatikk |
| 4 | Backend 2 | Seniorutvikler | 2 år | Ett prosjekt, applikasjon for ansettelse | Siviløkonom og elektroingeniør |
| 5 | Utleid/Back 1 | Seniorrådgiver/ seniorkonsulent | 3 år | Ett større prosjekt ute hos kunde, portal. | Master i Informasjons-systemer |
| 6 | Utleid/Back 2 | Utvikler | 6 år | Ett større prosjekt ute hos kunde, portal. | Master i Informasjons-systemer |
| 7 | Rådgiver | Prosjektleder og rådgiver. | 2,5 år | Ikke på fast prosjekt, skriver tilbud og driver rådgivning for øyeblikket. | Bachelor IT ingeniør, scrum-master og "certified scrum professional". |
| 8 | Avdelingsleder | Avdelingsleder, jobber også som tester og prosjektleder når nødvendig, men ikke for øyeblikket. | 11 år | Ett prosjekt utenom ledelse; Testoppdrag | Bachelor i IT-utvikling |
| 9 | Avdelingsleder/ Prosjektleder | Avdelingsleder, men jobber også med prosjektledelse, salg og rådgivning | 2 år | Flere prosjekter, mest på et større prosjekt med intranett- og internettportaler. | 4 år med systemering av databehandling |
| 10 | Leder | Leder og salgsansvarlig | 6 år | Hovedansvar for samtlige prosjekt | Bachelor i strategi og forvaltning, datafag og administrasjon og ledelse. |

Selskapet som ble kjøpt opp og som legger grunnlaget for min case studie ettersom hele avdelingen er hentet herfra, kan beskrives som et lite til middels stort IT-selskap med ulike kontorer for IT-drift samt et konsulent- og løsningsselskap med kontorer på flere lokasjoner i landet. Respondentene jeg har intervjuet kom samlet fra ett kontor i konsulentdelen av firma.

Det oppkjøpte selskapet hadde et stort antall små og mellomstore kunder i både privat og offentlig sektor i sitt lokalområde og i oppkjøpet fikk man dermed inkludert en større lokal markedsandel, noe som også til dels var hensikten med oppkjøpet. På denne måten får et av landets største konsulentselskap lokal kundenærhet som gir bedre evne til å forstå kundens virksomhet. De snakker dialekten, de kjenner behovene og de kan relativt raskt påta seg jobber og endringer og samarbeider tett med kunden. Med dette oppkjøpet håper man å kunne innta en rolle hvor man kan være en av de få totalleverandørene av IT-tjenester i området.

Med oppkjøpet i 2012 og offisiell overgang samt bytte av lokaler i årsskiftet 2012/2013 så fremsto avdelingen jeg har studert som en arbeidsplass i forandring. Mange ting endret seg, både praktiske ting som lokaler og systemer i bruk, men også endringer i organisasjon og hvordan man driver ledelse.

5 Resultater

Jeg vil i dette kapittelet beskrive funnene fra mine undersøkelser. Jeg vil først presentere hva respondentene hadde å si om testrutinene med dens kategorier, deretter presenterer jeg rutine for smidig systemutvikling. Mot slutten av kapittelet vil jeg analysere funnene mine inndelt i de to rutinene, samt at jeg også vil se på de kontekstuelle elementene.

5.1 Testrutine

Testrutinen i bedriften beskrives i liten grad som noe definert rutine som alle følger, men mer som noe alle gjør på den måten de føler er nødvendig for sitt arbeid. Noen likheter fremstår likevel i hvordan de behandler de ulike elementene rundt en testrutine og jeg vil her forsøke å presentere hva de forskjellige respondentene sa om disse elementene.

I min undersøkelse av testrutine valgte jeg å fokusere på åtte tema som til dels ble tatt opp av meg gjennom intervjuguiden og til dels ble nevnt av flere av respondentene. Jeg vil nå gå gjennom disse temaene og presentere hva de ulike respondentene hadde å si om dette.

Utviklerne som jobbet med frontend og design hadde relativt lite detaljerte beskrivelser av hvordan de gjennomførte testrutiner. Den ene utvikleren beskrev det slik:

”For min del så ser jeg det meste på skjermen med en gang og kan justere deretter med CSS eller HTML eller hva det skulle være og så be om kommentar fra andre når jeg egentlig føler det er greit for å se om de kan skrive under på det.” (Frontend 1)

5.1.1 Bruk av testplan

Testplan beskrives av frontendutviklerne som noe de ikke kjenner godt til eller bruker i stor grad. Dette eksemplifiseres videre av Frontend 1 *”... for min del så har jeg ikke brukt det. For min del er det mer at man har en konkret oppgave man skal løse og så skal man jo få det til å se ut sånn som man ble enige om i utgangspunktet” (Frontend 1).*

De fire respondentene fra ledelsen beskriver hvordan testplaner finnes i organisasjonen og at det dermed eksisterer standarder for dette som i utgangspunktet skal brukes. Likevel er testplan noe som er forbeholdt større prosjekter og er ikke noe de bruker til daglig på de mindre prosjektene. På spørsmål om det i utgangspunktet finnes en testplan fra øvre ledelse i organisasjonen får jeg blant annet svaret; *”i [bedriften], det er jeg temmelig sikker på” (Avdelingsleder/Prosjektleder).*

At testplan ikke brukes i utstrakt grad forklares av respondentene fra ledelsen med at prosjektene de vanligvis jobber med er av liten størrelse og lav kompleksitet og at testplan dermed ikke er nødvendig. Rådgiver beskriver at man høyere opp i organisasjonen har mer planer og rutiner for test når prosjektene er større:

”Mja, jeg tror at de virkelig store prosjektene gjør det, de feller det mer ned i den generelle prosjektplanen. At nå har vi en testperiode, nå skal vi gjøre det sånn og sånn og sånn. Og så har man dedikerte testledere som sørger for at de rette folkene jobber med det og man har testteam og sånt noe. Men det er mer for store prosjekter og sånt” (Rådgiver).

For respondentene som er utleid til annen bedrift beskrives det at de der har en testleder og at det er denne personen som har ansvar for å lage testplanen.

I tillegg beskrives det av Avdelingsleder samt Leder at man har eksempler på testplaner tilgjengelig slik at man kan bruke disse i prosjekter hvor det er nødvendig eller påkrevd. Da brukes disse som utgangspunkt for å utforme det man trenger.

”Men det vi har er jo eksempler på testplaner som vi har brukt hos ulike kunder. Så vi har på en måte eksempelplaner som da inneholder mye av dette, hvilke tester skal kjøres, hvilke ressurser skal være med, hvilket nivå skal du ha på testen etc.”
(Avdelingsleder).

5.1.2 Når det testes første gang

Den aller første testingen skjer så snart man har noe kode som enten kan fungere som ventet eller ikke. Dette er da hva man kaller utviklers test som ikke inngår som en egen del av den offisielle teststrategien, men som er mer en slags kvalitetssikring av koden som utvikler gjør selv. Utover dette beskrives det at testing skjer *”... fortløpende. Så fort vi har noe vi mener kunden kan begynne å få se på så tester vi selv først og så...”* (Backend 1).

Utenom de små testene som utviklerne gjør så tester utviklerne eller andre i teamet første gang når koden gir noe funksjonalitet som utviklerne anser som ferdig nok til at man kan klikke seg gjennom. Gjerne i henhold til spesifikasjonene. For de utleide utviklerne som har egen testperson som jobber på prosjektet så beskrives det slik;

”En testperson da, han vi jo ikke kunne teste noe før noe er faktisk så bra at det er i henhold til spesifikasjonen. Så vi kan si at, ser dette her greit ut, denne her lille biten? Så kan han si ja eller nei, men der er det stort sett at han vil ikke se på det før det er, altså dette er godkjent eller ikke godkjent for leveranse” (Utleid/Back 2).

Avdelingsleder som har erfaring som tester forteller også at man som testansvarlig gjerne starter allerede før kodingen med å skrive testcase, noe som ofte gjøres i samråd med kunden om mulig. Dersom kunden ikke har tid eller ønske om å delta på dette er det ønskelig at de i det minste ser gjennom og godkjenner testcasene.

”Altså, hvert fall sånn som jeg liker å jobbe så skal de involveres i forhold til å minimum godkjenne testcaser og det som forteller steg for steg hva man skal teste og et forventat resultat slik at de på en måte er med å godkjenner testcase/testscript”
(Avdelingsleder).

5.1.3 Ulike typer test

På spørsmål om hvilke typer test respondentene bruker svarer alle utviklerne som jobber med backend, både de utleide og de som sitter på avdelingen, at unit-test, eller enhetstest på norsk, brukes mye når de jobber. Dette er små testscript som utviklerne bruker for å kvalitetssikre koden, men kan ikke defineres som direkte inngående i en teststrategi.

”... vi lager mye unit-tester som vi koder/lager, for å sørge for at vi ikke driter oss ut. Vi har fryktelig mye kode som vi vedlikeholder og da er det, gjør vi en endring, før vi sjekker inn noe kode, så kjører vi testene” (Utleid/Back 2).

I tillegg til enhetstest bruker også utviklerne det man kaller integrasjonstest. De to utleide respondentene som beskriver test i sitt prosjekt forteller at dette gjennomføres automatisk en gang i døgnet for å sjekke at den nye koden passer sammen med resten av systemdelene. Integrasjonstest beskrives også av en av backend-utviklerne:

”Så da tar man jo webpart for webpart, får de til å fungere hver for seg, så man ser at alt går sammen og så begynner man å måtte sy dette sammen til slutt da. Og da blir det jo ny, testrutinen blir jo på en måte den samme, for å sjekke, for det første at syntaksen fungerer greit, at det ikke blir noe konflikt mellom webpartene, at det ikke blir noe konflikt med sånn som sharepoint er konfigurert” (Backend 2).

Bruk av akseptansetest, også kalt kundetest, beskrives av både backendutviklerne og de fire respondentene fra ledelsen. Dette er en test som kjøres av prosjektleder eller andre i ledelsen hvor det beskrives at *”så kjører man testen mot kunden og kunden skal akseptere på akseptansetesten sin”* (Avdelingsleder/Prosjektleder). En av utviklerne beskriver det i noe mer detalj med å forklare at de i *”... første levering i et prosjekt, da er det full akseptansetest og gjennomgang og det skal opp og fungere på serveren før man åpner opp for resten av verden og så videre”* (Backend 1).

5.1.4 Loggføring av test og resultater

De fire utviklerne som sitter og jobber med prosjekter for avdelingen sier alle at de ikke er like flinke til å logge testing og resultater som de kunne vært. Det fremgår ikke noen klare eller formelle krav fra ledelsen, verken i avdelingen eller konsernet, til omfattende logging. Likevel er utviklerne tydeligvis klar over at det hadde vært å foretrekke om de logget mer. Noen melder at de noterer litt for eget bruk, men at man for det meste løser feilene som dukker opp i testene med en gang, i stedet for bruk av loggføring og kategorisering.

De to utviklerne som sitter utleid til annet firma jobber under mer rigide testrutiner og begge melder fast bruk av systemet Team Foundation System (TFS) for å logge tester og resultater. Redmine er et system som blir nevnt som noe man brukte tidligere, det vil si før oppkjøpet, men som nå er på vei ut til fordel for TFS.

Tre av fire respondenter fra ledelsen beskriver også bruk av TFS for loggføring i forbindelse med testrutine. Redmine beskrives av to fra ledelsen i tillegg til en av de utleide respondentene, men i mindre grad da det er utbredt enighet om at dette er et system som er på vei ut. Avdelingen ønsker heller å gå over på TFS da dette er et system som er mer i tråd med satsningen til avdelingen.

”Og det [Redmine] funka veldig bra på en del sånne småprosjekter. Men det vi ønsker er jo å forholde oss til TFS. Det er jo på en måte Microsoft sin. Man får på en måte mye ekstra. Hvis man legger opp løpet ordentlig i TFS så får man mye ekstra.. Validering, sjekking automatisk av hva som skjer og sånne ting, det får man ikke i Redmine på samme måte” (Leder).

5.1.5 Når kunden involveres i test

Fire av seks utviklere er enige i at kunden involveres så snart man har en funksjonalitet som man har testet gjennom selv og som man kan anse som ferdig og i henhold til spesifikasjon. I noen prosjekter vil det si mot slutten når man har ferdigstilt hele systemet dersom det er et

relativt lite prosjekt. I andre situasjoner hvor man kan bruke delleveranser så vil man forsøke å involvere kunden for å teste flere ganger i løpet av prosjektet.

Avdelingslederen som har erfaring som tester beskriver at kunden aller helst burde involveres allerede før man starter kodingen slik at de kan delta i skrivingen av testcase som beskrevet i kapittel 5.1.2. En av de utleide utviklerne samt den andre avdelingslederen beskriver mer hvordan kundeinvolvering varierer i forhold til hva kunden ønsker. Noen vil delta hele tiden og ha kontinuerlig kontroll i motsetning til andre som kun er interessert i å teste et ferdig produkt.

”Noen prosjekter som jeg har vært inni, der har kunden bidratt med egen testing. Så de har da engasjert seg egentlig helt fra starten av, nesten vært tilgjengelig hos oss. I andre prosjekter så har kunden ikke testa noe før vi har hatt en ferdig leveranse nesten, så har de gått gjennom på siste sjekk og sett at ok, dette er bra og så testa da” (Utleid/Back 1).

5.1.6 Likhet i testrutine fra prosjekt til prosjekt

Tre av fire respondenter fra ledelsen er klare på at testrutinen er ulik fra prosjekt til prosjekt og at dette er grunnet kundens ønsker samt størrelse og kompleksitet i prosjektet. Kunden fordi de bestemmer selv hvor involvert de ønsker å være i utviklingen og testingen av prosjektene. Størrelse påvirker rutineene fordi store prosjekter krever ofte mer testing enn mindre prosjekter og med større prosjekter bruker man gjerne mer standardiserte rutiner både for testing, utvikling og prosjektadministrasjon.

Den fjerde respondente fra ledelsen beskriver først hvordan testrutinene er like og sier sin mening om kvaliteten på disse rutineene.

”Jeg vil ikke si at rutineene er dårlige, det er de ikke, de fungerer, men de kunne ha fungert mye bedre og vært satt i system. Altså i et mer rigid system, da hadde de klart å isolere folkene til oppgavene sine mye mer. Og de er like i alle prosjekter egentlig” (Rådgiver).

Senere i intervjuet beskriver han mer hvordan det likevel er forskjeller mellom prosjektene.

”Men inni der så finnes det jo en masse rutiner som man gjør som er forskjellig fra prosjekt til prosjekt. Og det er jo forskjellig fra folk til folk også, hvordan samarbeider de og hvordan de deler informasjon...” (Rådgiver).

Utviklerne gir også mer tvetydige svar på hvorvidt det er likheter i testrutineene. Fem av seks utviklere beskriver at det er forskjeller mellom prosjektene og at man ikke egentlig kan sammenligne et prosjekt med et annet. Likevel ytres det av tre av de samme utviklerne at det er likheter i prosjekter av samme størrelse eller hvor man gjør lignende oppgaver. Den siste utvikleren svarer på spørsmål om variasjon i testrutine at; *”nei, det er noenlunde det samme”* (Backend 1).

5.1.7 Kodegjennomgang - andre tester din kode

Dette er et konsept som ble nevnt alle utviklerne som jobber inne på avdelingen som en grei løsning for å sikre at man ikke overser feil eller problemer i egen kode. De fleste utviklerne, både frontend og backend beskriver at de ofte skriver kode, tester selv om det fungerer som

ventet, samt at det ser ut som de ønsker. Deretter ber de gjerne en annen person på teamet om å ”klikke gjennom” løsningen for å se om de har samme opplevelse. Da er det ofte et poeng at det er en prosjektleder eller en annen som ikke utvikler selv som ser på løsningen.

”... prosjektleder eller en som ikke har gjort programmeringsjobben som trykker gjennom og tester når det ligger på test og så går det over til kunden hvor han trykker gjennom og ser at det funker” (Backend 1).

Det var kun én av de utleide utviklerne som var uenig i utsagnet om gjennomgang av andres kode. Han beskriver at det forekom at han kikket på hva de andre på teamet gjorde av nysgjerrighet, men det var svært sjeldent at han ba andre om å teste sin kode. Han mente at dette var noe man gjerne gjorde mer når man var ny og usikker.

”Av og til så har man vel gjort det, i alle fall de første årene, når man ikke helt hadde peiling på hva man holdt på med kanskje.. Så gjorde en kanskje det, og det er jo en veldig grei måte som junior i alle fall, å kunne få litt støtte på hva en driver med, med en code review, men jeg har ikke sett noe formalisert praksis rundt det” (Utleid/Back 2).

Begge avdelingslederne beskriver også gjennomgang av andres kode som et godt verktøy for å kvalitetssikre arbeidet. En nevner spesifikt at han ønsket å ha mer tid til denne typen test; *”... jeg prøver, men der kan man også bli bedre, å få tid til det. Jeg prøver å gå gjennom, hvert fall av de prosjektene som jeg er delaktig i...” (Avdelingsleder/Prosjektleder).*

5.1.8 Kriterier for godkjenning og kategorier av feil

Kun halvparten av respondentene svarte på dette og de beskrev at godkjenning av test var opp til kunden og at det var de som godkjente eller nedkjente til slutt. Ble testen godkjent så ville løsningen kjøres ut i produksjon og dersom kunden fant feil teamet ikke hadde sett selv, så ble det endret og testet på nytt. I den grad det ble beskrevet hvordan testere drev godkjenning av tester med satte kriterier så var dette i forhold til en kravspesifikasjon.

Kategorisering av feil som oppdages blir gjort i liten grad når utviklere tester egen kode eller andres, men er mer noe som brukes av testledere eller andre som tester i en mer formalisert setting. Leder og Utleid 2 beskriver kategoriene business critical for når hele systemet eller funksjonaliteten stopper samt high, medium og low. Utleid 1 går ikke inn i detalj, men forklarer at kriterier og kategorier utvikles og håndteres av testleder i prosjektet han sitter på nå.

5.2 Rutine for smidig systemutvikling

Systemutvikling kan gjøres på mange måter, men i denne bedriften kommer det klart frem et ønske om å være smidige og denne smidigheten beskrives som et utspring fra metoden Scrum. Mine spørsmål ble stilt for å finne ut hva det er som gjør bedriften smidig. Dette forsøkte jeg å kartlegge ved å stille spørsmål om hvordan de velger å gjøre ting som kvalitetssikring, møtevirkosomhet, arbeidsfordeling og leveranser samt flere andre elementer, både tatt opp av meg og nevnt av respondentene. Ikke for å se hvorvidt bedriften faktisk bruker Scrum eller ei, men for å se om de kan defineres som smidige og om praksisen tilsvarer hva bedriften egentlig ønsker å gjøre. En av respondentene beskrev smidigheten til bedriften på denne måten:

”Gitt at definisjonen på smidig er evnen til å snu seg fort og assimilere ny informasjon raskt og gjøre slutninger basert på erfaring så vil jeg si at [bedriften] er et sånn noenlunde smidig selskap” (Rådgiver).

5.2.1 Arbeidsfordeling

Tre av utviklerne som ikke er utleid er klare på at oppgavene i de forskjellige prosjektene fordeles i stor grad av prosjektleder. Leder beskriver at det i noen tilfeller brukes brukerhistorier, men at dette ikke er normen. Alle utviklerne i avdelingen beskriver at de fordeler oppgaver på statusmøtene og at disse ofte deles basert på evne og erfaring. *”Så det er litt sånn at hvis man kan det, så får du oppgaven”* (Backend 1). De to utleide utviklerne beskriver bruk av sprint planning, estimering og brukerhistorier i sitt prosjekt ute hos kunde.

Når det gjelder estimering og prioritering av oppgaver i prosjektet så beskrives det av tre av utviklerne at dette ofte gjøres av prosjektleder, men at når utviklerne skal begynne å jobbe så hender det at rekkefølgen må endres litt. En av backendutviklerne gir et eksempel:

”Men hvis jeg ser at den webparten, for at den skal funke helt optimalt, at man skal kunne se at den fungerer, så er den avhengig av nummer to.. Så derfor så vil gjerne jeg velge å begynne på nummer to i stedet for” (Backend 2).

En av frontendutviklerne beskriver i motsetning til de andre, hvordan det er utviklerne som estimerer oppgavene i prosjekter: *”Det er vi selv som i hovedsak estimerer, og så gjør jo prosjektleder en sammenstilling av estimat og tar en vurdering av hvilke tilbud som skal gis”* (Frontend 2).

5.2.2 Møtevirksomhet

De to utleide utviklerne beskriver en mye større grad av møtevirksomhet enn de resterende respondentene. De utleide bruker møteelementer fra Scrum som daily standup, sprint planning og sprint retrospective. Møtevirksomheten som brukes på avdelingen beskrives av halvparten av respondentene som stort sett begrenset til et ukentlig statusmøte/prosjektmøte hvor teamet med utviklere snakker sammen om status på de ulike prosjektene som jobbes med.

Under observasjon av et slikt møte så jeg at hensikten var å etablere status på de forskjellige utviklerne i forhold til de ulike kundene og prosjektene. Tilstede på møtet var avdelingsleder/prosjektleder som ledet møtet samt fem utviklere.

Møtet var relativt løst satt opp hvor deltakerne satt avslappet med kaffekopper og diskuterte i felleskap hvordan de ulike prosjektene gikk. Kontrollen av prosjektene og kundene foregikk med enkelt excel-ark som viste de ulike prosjektene og oppgavene innenfor hvert prosjekt, med inndeling for hvem som jobbet på hva og hvor mange timer som gjenstod på oppgaven. Prosjektene var videre merket med fargekodene grønt, (alt alt ligger bra an), gult (fortsatt noe å gjøre, men ikke hastverk), oransje (en del gjenstående arbeid som begynner å haste) og rødt (krise, her brenner det, man må få gjort noe snarest). Planen fremover var visstnok å få utbedret dette dokumentet til et noe mer organisert styringsverktøy (prosjekttrom). Foreløpig fungerer excel som et dynamisk dokument som teamet har tilgang på gjennom intranettet hvor det oppdateres og endres etter behov.

Under møtet ble dokumentet oppdatert alt ettersom hva utviklerne hadde å si om hvor langt de var kommet. Under samtalene om hvordan de ulike prosjektene gikk ble det også tidvis

diskutert svært tekniske spørsmål om hvordan utviklerne best kunne løse problemer eller utfordringer i arbeidet. Det ble også til dels snakket om testing, hvor avdelingsleder tilbyr seg å teste noen av funksjonene. Av andre tema som dukket opp i samtalen var mangelen på folk. Teamet fremstår som generelt sett travle og med tanke på oppkjøpet og at noen har sagt opp har de behov for flere ansatte. Mange av prosjektene er merket med rødt og det blir nevnt flere ganger at utviklerne jobber mye med å ”slukke branner” fremfor å jobbe jevnt med ett og ett prosjekt av gangen. Det blir nevnt at det kommer en ny ansatt i april som de fleste ser frem til, det ble under møtet også til dels diskutert hvordan denne ansatte burde brukes for å få mest mulig nytte.

I tillegg til statusmøte/prosjektmøte beskrives det av flere utviklere at man ofte tar en samtale ”over bordet” i stedet for å kalle inn til offisielle møter for å diskutere arbeid. På denne måten får teamet diskutert enkle ting på en rask og effektiv måte uten å måtte booke møterom og sende innkallelse.

”Man har avklaringsmøte med kunden, der de er nødt for å ha med en utvikler, så er det interne prosjektmøte, det vil si bare de som er involvert i prosjektet, det kan være at utviklerne er nødt for å bare snakke sammen og det tar vi jo veldig ofte bare rundt bordet. Hvis vi er tre stk som trenger å konferere litt så setter vi en tid og så bare sitter vi på en maskin hvor vi har alt sammen oppe” (Backend 2).

Som man ser fra Backend 2 så avholdes også andre møter med kunde og lignende, men disse ble ikke beskrevet i noen utstrakt grad.

5.2.3 Fremtidig prosjekttrom

Dette er et av konseptene som tidlig ble brakt opp av respondentene. Det er ikke et element fra smidig utvikling eller noe man har fra Scrum, men det er et verktøy som ønskes innført. Ideen kommer først og fremst fra en av avdelingslederne som har erfaring med et lignende konsept fra et tidligere firma han jobbet for. Det han ønsker seg for avdelingen er:

”... et prosjekttrom for hver kunde vi har og hvert av de prosjekttrommene, der skal kontrakt/avtale ligge, der skal ligge liksom oppgavelister og tasklister og sånne ting og det er når vi gjennomfører prosjektene så skal den lista ligge der. ... så vil jeg kjøre en test da på oppgavene hvor jeg har liksom sånn typisk status, arbeidsflyt, ny, til analyse, analysen skal si om den er designa eller utvikla, utvikling, da skal den jo gå til intern test og den er testa ok eller denne feila ikke sant. Og så er det da til kundetest og så videre” (Avdelingsleder/Prosjektleder).

Dette prosjekttrommet ønsker han å lage i Microsoft SharePoint som er det verktøyet som brukes for store deler av utviklingen i avdelingen. Kundene skal ha tilgang til systemet og de skal selv kunne melde inn saker i sine egne prosjekttrom og skal kunne se hva som blir gjort og fremgangen i prosjektet.

Systemet nevnes av halvparten av respondentene, men to av disse uttrykker klart noe skepsis til om systemet noen gang kommer til å settes i drift da det ikke er ferdig utviklet enda. Intervjuet med Avdelingsleder/Prosjektleder som er champion for systemet, fant sted tidlig i mars og etter spørsmål på e-post i mai får jeg beskjed om at systemet fremdeles ikke er i drift. En av skeptikerne, en av de utleide utviklerne beskrev tvilen slik:

”Ja, men det skorter jo ikke av forslag og tiltak. Men det er gjennomføringen av det som er vanskelig. Som hver kunde er forskjellig så er, jeg tror det er veldig typisk i

konsulentbransjen. Hvor man har mange prosjekter og mange kunder så klarer man ikke å få denne felles prosjektmetodikken eller felles måte å gjøre ting på. Det er veldig mange gode ideer og veldig mange gode tiltak som blir løfta og tatt frem og kanskje brukt masse penger på, men så forsvinner det igjen” (Utleid/Back 1).

5.2.4 Endringsønsker og uforutsette hendelser

Endringsønsker fra kunde og plutselige problemer beskrives av alle utviklerne i avdelingen som noe som skal håndteres av prosjektleder. Dette er derimot ikke alltid slik det skjer og det beskrives videre av to av disse, som har nær kontakt med kunden, at de ofte kontakter utviklerne direkte i håp om å få løst problemet raskere. Leder av avdelingen beskriver hvordan han ønsker å skjerme utviklerne i så stor grad som mulig for denne type kontakt; ”... da prøver vi å verne utvikleren sånn at ikke han må sitte og drive og diskutere dette her fordi han har nok med å gjøre det han faktisk skal gjøre” (Leder).

På tross av at man gjerne forsøker å unngå direkte problemløsning mellom kunde og utvikler, så beskrives det av tre utviklere at dersom det er et lite problem som har en relativt enkel løsning, så vil feil eller små endringer bare fikses så snart som mulig.

I intervjuene med ledelse kommer det i tillegg frem at endringsønsker og feilretting gjerne er avtafefestet og at kunden får et estimat på hva det vil koste i ekstra tid og penger å få løst problemene. I tillegg har man kunder som har drifteavtaler og da må man gjerne se på denne for å avgjøre hvor snart man må gjøre endringen eller løse problemet.

De uleide utviklerne beskriver bruk av serviceavtale og testleder i tillegg til prosjektleder som metode for håndtering av endringsønsker og uforutsette hendelser. Men også her svarer en av utviklerne at dersom han sitter hos en kunde og jobber på et prosjekt og man får beskjed om at; ”nå er det noe som har gått til helsikke, og de sier slipp det du har i hendene og fiks det, så ville jeg nok tatt det” (Utleid/Back 2).

5.2.5 Leveranser

I utgangspunktet ligger gjerne konseptet inkrementelle leveranser i måten man jobber smidig på, og i noen prosjekter beskrives det at man bruker delleveranser. Likevel beskrives leveranser av samtlige respondenter som noe svært varierende i forhold til prosjektet. Noen prosjekter er så pass små at de kun har én leveranse på slutten av prosjektet, mens de to utleide utviklerne som sitter på et stort prosjekt for kunde beskriver mer bruk av inkrementelle leveranser av små deler. På spørsmål om det brukes delleveranser så svarer en av backendutviklerne slik:

”Ja, på nye prosjekter så hender at vi gjør det. Prosjekter som jeg har vært på har liksom ikke vært store nok. Hvis kunden sier at denne funksjonaliteten må vi ha mens det er noe annet som han gjerne vil ha, så leverer vi gjerne det de må ha først så de kan begynne å bruke det litt” (Backend 1).

5.2.6 Evaluering/kvalitetssikring

Evaluering av gjennomførte prosjekter er noe som nevnes spesifikt av to i ledelsen som noe man burde vært flinkere til:

”Og så når du er i produksjon så er det vel etter release, det er der som vi skifter litt på, men det er jo da ønskelig at en gjør en evaluering sammen med kunden om hva er det vi egentlig har, kunden og hvor fornøyde de egentlig har vært. Men der har vi en sånn minus med at vi er ikke flinke nok til det enda” (Avdelingsleder/Prosjektleder).

Kvalitetssikring beskrives av alle respondentene bortsett fra de utleide utviklerne som noe som gjøres i liten grad eller ikke i det hele tatt. I den grad noe kvalitetssikring gjennomføres så er dette begrenset til testing. De to utleide utviklerne beskriver at man bruker egne kvalitetssikringsmiljø på sitt prosjekt og at dette er nært knyttet til testleder.

I utgangspunktet er konsultentselskapet ISO-sertifisert på kvalitetssikring og det er meningen at man skal ha relativt strenge rutiner. Likevel er dette noe som flere av respondentene ikke har hørt om og på langt nær noe som påvirker hvordan de gjør kvalitetssikring. Dette forklares av leder med at det er vanskelig å bruke den type stor metodikk på såpass små prosjekt. Han sier videre at de fleste prosjektene de gjennomfører har lav risiko og at man dermed ikke ønsker å bruke mye tid og penger på sterk kvalitetssikring.

5.2.7 Hva gjør bedriften smidig?

På spørsmål til respondentene om hva de mener gjør avdelingen smidig, fikk jeg en rekke ulike svar. To i ledelsen svarte tydelig at det var menneskene i teamet som utgjorde forskjellen her. De ble beskrevet som skarpe og flinke, erfarne og vet hva de holder på med samt at de ble beskrevet som sosialt intelligente, det vil si at de var flinke til å stille spørsmål i vanskelige situasjoner og man kunne dermed enklere overlate mer ansvar til de ansatte.

De to andre i ledelsen beskrev avdelingens evne til å håndtere endringer raskt og muligheten til å dele opp i mindre oppgaver som nøkkelen til avdelingens smidighet. En av de utleide utviklerne, i samtale om hvordan han oppfatter smidighet i avdelingen hvor han er ansatt, ikke utleid til, beskriver det slik:

”Altså det som gjør oss smidige, hvert fall oss her i [avdelingen] er at vi har ei samling av mye skarpe folk med forskjellig bakgrunn og forskjellig kompetanse som gjør at vi kan omstille oss veldig fort til ulike prosjekter. Altså den gruppa med mennesker vi har her kan ta unna veldig mye forskjellig” (Utleid/Back 2).

To av utviklerne som sitter i avdelingen beskriver god kommunikasjonsevne og størrelse på prosjektene som medvirkende til bedriftens smidighet.

5.2.8 Hvem legger føringene?

Av de syv respondentene som svarte på dette spørsmålet var det mange delte meninger. Leder og Avdelingsleder var ganske tydelige på at det var dem, altså bedriften som først og fremst la føringer for hvordan arbeidet skulle gjøres. De nevner de fagansvarlige, prosjektledere og avdelingsledere som de som legger føringer. Avdelingsleder svarer slik på spørsmålet:

”Det er jo sånn sett oss kan du si, men hvem er oss her? De som sånn sett har et visst fagansvar og prosjektledere ikke sant, vi har jo kjørt prosjektlederkurs internt på våre når vi var i [bedrift før oppkjøp], sånn at vi kan prøve å bli litt mer tydelige på at vi gjør ting likt. Så man blir på en måte enige om noen fellesting. På hvordan man skal gjøre det” (Avdelingsleder).

Likevel var to av utviklerne og en av prosjektlederne enige om at det stort sett er kunden som bestemmer hvordan man skal jobbe. Til slutt mente en av de utleide utviklerne klart at det verken var prosjektleder eller kunden som bestemte hvordan man jobber, men heller at det var utvikler som vanligvis la føringene. Han beskrev dette slik:

”... vi legger veldig føringen på hvordan vi ønsker å jobbe for at vi skal kunne prestere. Og det er jo ofte der i prosjektene går, at det vi leverer, som kunden ser. Og for at vi skal levere det kunden forventer eller det som selgeren har solgt inn, så må det i mange tilfeller skje på våre premisser” (Utleid/Back 2).

5.3 Analyse av data

I dette underkapittelet vil jeg forsøke å analysere resultatene jeg fikk gjennom mine intervjuer for å gi et overblikk av de to rutineene og hvordan de forholder seg. Jeg vil følge oppsettet jeg har brukt i beskrivelsen av resultatene slik at det skal bli oversiktlig og lett å følge.

5.3.1 Testrutine

Bruk av testplan var et av de overordnede konseptene som jeg forventet ville være i utstrakt bruk og noe som ville diktere hvordan utviklerne og andre i teamet måtte jobbe og teste i forhold til øvre ledelse. Jeg ventet at testplan skulle være ”regelboken” for hvordan testing i prosjektene skulle utføres og at denne ville være endres ettersom tiden gikk og prosjektene utviklet seg. I realiteten viste det seg at fire av seks utviklere verken kjente til detaljene i en testplan eller at de brukte den noe særlig. På spørsmål om det fantes en testplan fikk jeg svaret; *”Det må du spørre de som er dypere i kode tror jeg. De som driver med ren utvikling”* (Frontend 1). Videre spurte jeg hva innholdet i en testplan er, da fikk jeg eksempelvis svaret; *”Det er ikke jeg riktig person til å svare på”* (Utleid/Back 1). Et siste problem med bruk av testplan er at det insinueres at omfattende testing er dyrt. Dyrere enn hva de lokale kundene ønsker å betale. *”Her er vi jo mye mer sentrert rundt SMB-kundene, det er de som misliker at du bruker 10 000 kr på noe”* (Rådgiver).

I likhet med forventningen om at virksomheten ville ha en overordnet testplan i bruk hadde jeg også en forventning om at man hadde et mer generelt sett med rutiner og arbeidsprinsipper som kom fra øverste ledelse og at det var ventet og ønsket at hele konsernet jobbet under de samme reglene. Dette viste seg raskt å være en feil antagelse fra min side. Det kom frem allerede i samtale med utviklerne som ikke ga uttrykk for å være påvirket av konsernledelsen i det hele tatt, men viste seg enda klarere da jeg snakket med lederen for teamet. Han sa rett ut at han jobbet mye med å skjerme utviklerne fra toppledelsen og at han ønsket at teamet skulle kunne fortsette å jobbe slik de var vant med fra før oppkjøpet.

På spørsmål til leder om teamet må belage seg på forandringer i mengden overhead og retningslinjer fra konsernledelsen får jeg svaret:

”Nei, det gjør vi bevisst. For at de skal på en måte levere det de er flinke til og det å jobbe som utviklere. Gi kundene våre råd, gi dem best mulig løsninger. Det er det de er flinke til. Så skal vi andre sørge for at vi holder oss til konsernets retningslinjer og på en måte rapportere på den måten vi skal rapportere på og at vi leverer på en måte i henhold til konsernets forventninger... Så de skal på en måte ikke bli belemra med interne problemstillinger eller sånne ting” (Leder).

I forbindelse med når det testes første gang kom det frem, til en viss grad mellom linjene, at utviklerne selv var så pass erfarne og dyktige i sin jobb at de visste hvordan å drive tidlig testing for å sikre god kvalitet i koden ved bruk av enhetstester og lignende. I tillegg beskrev de hvordan utviklerne ofte bedømte selv når noe var 'ferdig nok' for å vise til kunde; *"Så fort vi har noe vi mener kunden kan begynne å se på"* (Backend 1).

En av respondentene fra ledelsen beskriver også hvordan han mener at utviklerne selv i stor grad har ansvaret for testing av egen kode:

"Du som utvikler har ansvar for hva jeg kaller 'good engineering practices', du skal sørge for at du leverer god kvalitet. Du har ansvar for den biten av kode som du har skrevet helt til den kjører i produksjon" (Rådgiver).

Alle utviklerne som jobber med backend nevner en rekke ulike typer tester, men de aller vanligste fremstår tydelig som enhetstest som man gjør selv, eventuelt en integrasjonstest med de andre systemene eller bestanddelene, samt akseptansetest som brukes ut mot kunden. Også ledelsen beskriver at det ofte er denne typen testing som utføres.

"Så tester man hver enkelt kode, hver enkelt enhet, og så tester man samhandling og så tester man til slutt med akseptansetesting. Det er jo et sånt normalløp, hvor alle er happy. Men da skal man jo på prosjekter som er ganske store. Da snakker man jo om millionprosjekter. Og det er ikke ofte vi har her" (Leder).

Men som man også får inntrykk av fra dette sitatet så er de fleste prosjektene såpass små i antall ansatte og antall arbeidstimer at de ikke ser nødvendigheten av strenge testrutiner. Det ser ut til at de dermed velger bort faste metoder og strenge rutiner for hvordan de skal jobbe.

Loggføring av test og resultater var også en av de tingene som jeg hadde ventet ville være påkrevd i større grad og at man hadde standardiserte systemer og løsninger for dette. Det viste seg at ledelsen ønsker å bruke TFS som standard, men det fremstår likevel ikke som noe de påkrever utviklerne til å bruke når de tester. Dette kan være noe av det som bidrar til at utviklerne ikke dokumenterer noe særlig av testene sine. Dette beskrives videre av en av backendutviklerne; *"... så klart, da blir det litt mer sånn at hvis utviklere skal sitte å gjøre alt, da vet man jo alltid at dokumentasjonen uteblir. Det blir mangelfull dokumentasjon"* (Backend 2).

Det er vanskelig å sammenligne de ulike prosjektene da de beskrives så ulikt fra respondentene. Likevel sitter jeg igjen med et inntrykk av at det finnes en viss grad av likhet likevel og at det er en iboende rutine som (med noen variasjoner) brukes fra gang til gang. Jeg så også at selv om det kanskje ikke ble beskrevet én felles rutine av alle respondentene så var det likheter å finne i hva de individuelle utviklerne og lederne gjorde fra prosjekt til prosjekt. En av frontendutviklerne beskrev det slik:

"Men i forhold til design og implementering av design, interaksjonsdesign, så er det den samme prosessen mer eller mindre hver gang. Det er litt varierende grad av involvering av kunden fra starten. Avhengig av prosjektets størrelse og kundens betalingsvilje og egentlig involveringsvilje... Prosessen blir for så vidt den samme, men omfanget varierer" (Frontend 2).

Kodegjennomgang er ikke noe formalisert prosess og fremstår ikke som noe ledelsen er klart bevisste på, men det virker likevel som noe de fleste utviklerne velger å bruke selv, for å få friske øyne på det man jobber med. Det å 'klikke gjennom' noe som andre har kodet blir beskrevet som en teknikk for selvtest hvor man ber de rundt seg om en tjeneste. En av

frontendutviklerne beskriver hvordan han gjerne ber *”om kommentar fra andre når jeg egentlig føler det er greit, for å se om de kan skrive under på det”* (Frontend 1).

Kriterier for godkjenning av test beskrives svært enkelt av de fleste med uttrykk som; *”så gjør man noe, ser om det virker, gjør det om igjen når det ikke virker osv.”* (Frontend 2). Det fremgår at man ikke har faste kriterier som er satt opp og planlagt på forhånd, men at det heller er i retning av at det enten fungerer og man kan gå vider, eller så fungerer det ikke og man må gjøre noe annerledes.

Feil kategoriseres sjelden og de kategoriene som nevnes er forbeholdt bruk i større prosjekter med faste testere og testrutiner i mye større grad. Et eksempel nevnes av Leder:

”... det kommer jo litt an på hva slags avtale vi har med kunden. Mot [kunde] så har vi jo en stor forvaltningsavtale. De betaler jo rundt en 225 000 i måneden på en løsning vi har hos de. Og der er det jo klart lagt ned i avtalen, lagt ned en kategorisering på hva som er feil. Hva er liksom business-critical, hva er high, hva er medium, hva er low” (Leder).

For å oppsummere ser jeg at på tross av at respondentene i mange tilfeller ikke definerer det de gjør til daglig som noen rutine så kommer det frem en rekke overordnede likheter, som når man ser dem i en større sammenheng fremtrer som en slags rutine som avdelingen bruker. Det kan virke som denne rutinen er til en viss grad ubevisst og den fremstod for meg på følgende måte:

1. Skrive kode, teste denne selv med bruk av enhetstester.
2. Få noen andre til å kikke gjennom funksjonaliteten for å se om friske øyne ser noe utvikleren har gått glipp av. Gjennomtitt helst av prosjektleder eller andre som ikke programmerer selv.
3. Kjøre regresjonstest, se om det man har kodet passer med resten av systemene.
4. Legge det ut på testserver for offisiell testing dersom det er et større prosjekt og man har en testserver. Er prosjektet av liten størrelse går rutinen direkte til punkt 5.
5. Sette opp en test av systemet for kunden. Kunden skal gjøre en akseptansetest.
6. Fikse eventuelle problemer (eller forandringer) kunden finner.
7. Ny akseptansetest med kunden etter eventuelle endringer/forbedringer.
8. Rulle ut til produksjon.

5.3.2 Rutine for smidig systemutvikling

I utgangspunktet er rutinene i teamet basert på smidige metoder, nærmere bestemt; Scrum. Dette bekreftes av samtlige respondenter fra ledelsen i tillegg til tre av utviklerne. Rådgiver beskriver hvordan man bruker noen av elementene fra Scrum.

”Vi bruker jo Scrum, men det er jo vanskelig å følge det helt slavisk for ofte så jobber man på flere prosjekter sant, man kan ikke ha de der daglige standupene, det passer kanskje ikke med møte, når folk er tilgjengelig og alt mulig sånt noe. Men man går jo gjennom disse her, det som jeg kaller fysikken av Scrum” (Rådgiver).

Det er likevel forskjellige tolkninger av hvor smidig de faktisk er og i hvor stor grad de bruker Scrum. Scrumelementene er få og det er ikke mye av metoden som brukes jevnlig i teamet. Likevel holder de fleste på tanken om at de er smidig og de har noe forskjellige

meninger om hva det er som gjør dem til dette. De fleste forklarer mangelen på en mer definert metode med prosjektene og jobbens natur. Prosjektene er som oftest små, de har ikke behov for et helt scrum-team for å kunne gjennomføre et prosjekt og de klarer heller ikke isolere et scrum-team med bare ett prosjekt av gangen.

”Scrum er en kjempebra metodikk hvis man på en måte klarer å isolere utviklingsteamet sammen med kundeteamet sånn at de hele tiden jobber integrert hele tiden.. Men de fleste prosjektene vi snakker om her der har du, fra kundens side, så har man ressurser som kanskje er tildelt 20%, 40%, 60% prosjekttid, men de har ikke fått noe mindre oppgaver de skulle gjøre i sin normale arbeidsdag, så de har jo 100% til vanlig” (Leder).

I tillegg beskrives det av en av utviklerne at størrelsen på prosjektene i form av kostnader ikke er overensstemmende med metodebruk. De mener blant annet at de ikke klarer å forsvare bruk av et kvarter til morgenmøte hver dag til kunder som betaler dyrt for utviklernes tid.

I teamet jeg snakket med var noen av utviklerne utleid som konsulenter til et annet firma og disse kom spesifikt inn til kontoret for å bli intervjuet. Begge disse var utleid til samme firma og de beskrev ganske annerledes rutiner for sin arbeidshverdag. I eksempelprosjektet de snakket om, vitnet det om en mye klarere bruk av Scrum som metode. Her ble flere av elementene brukt og det ble gjennomført både morgenmøte, backlog, sprinter og brukerhistorier for å nevne noen.

”Vi har alle de tingene man kan se utenifra, fra Scrum, bruker vi. Men om de blir brukt på riktig måte det kan ikke jeg helt svare på fordi jeg ikke har analysert det så mye, men man kan nok si det sånn at vi ikke kjører Scrum. Det kan man si. En method-in-action eller cargo culture som andre ville kalle det” (Utleid/Back 1).

Videre forklarte de denne metodebruken (sett i forhold til hvordan de hadde gjort ting tidligere) som i stor grad mulig fordi det var et betraktelig større prosjekt med høyere kompleksitet, samt flere folk og mer penger. I tillegg var det et internt prosjekt og de er dermed ikke avhengig av en ekstern kunde som skal være tilgjengelig hele tiden.

At Leder beskriver lite bruk av brukerhistorier kom som en overraskelse. Jeg hadde forventet at denne avdelingen som beskriver seg selv som smidig og i mange tilfeller nevner Scrum spesifikt ville ha brukerhistorier. Prosjektledere velger i stedet å fordele arbeidsoppgaver, eventuelt så får utviklerne velge selv. En mulig forklaring på hvorfor man jobber slik kommer fra en av utviklerne som beskriver at *”noen ting ligner jo mye på ting man har gjort før, så da kan man bruke en del av det”* (Frontend 1). Leder beskriver videre hvordan man lar være å bruke elementer i Scrum som for eksempel Planning Poker:

”... vi ser at vi veldig ofte gjør de samme oppgavene hos forskjellige kunder sånn liksom, når man da har gjort samme oppgaven to ganger, så det å sitte å bruke tid på poker planning når man vet at det tar 4 timer” (Leder).

Møtevirksomheten i avdelingen fremstår i stor grad som behovsbasert og man har ikke flere møter enn man må. Som nevnt i kap. 5.2.2 så beskrives det ukentlige statusmøtet som det eneste ordinære møtet mellom utviklere og ledelse. Flere av utviklerne beskriver også at man gjerne unngår møter om mulig, ved heller å snakke ved arbeidsstasjonen eller såkalt ”over bordet”.

”... vi snakker jo mye over bordet og prøver å løse problemene smidig på den måten hvis man kan si det sånn. Diskuterer problemstillinger eller tar opp noe med

prosjektleder der og da, så man kan gjøre endringer i prosjektplanen eller ringe kunden eller gjøre aksjoner på det, så sånn sett så er vi jo smidige” (Frontend 2).

Planene om innføring av prosjekttrom er et interessant tema som viser hvordan man har en i ledelsen som virkelig brenner for å få innført dette, men likevel går innføringen sakte. Avdelingsleder/Prosjektleder som er champion for systemet beskriver at det er nesten ferdig programmert og at han regner med at det er innført om et par måneder. Ved kontakt to måneder senere er systemet fremdeles ikke innført.

Håndtering av endring og generelle uforutsette hendelser ser ut til å være et av områdene med størst variasjon i måten det håndteres på. Dette kan naturligvis ligge til dels i det faktum at uforutsette hendelser er nettopp det; uforutsette. Det er vanskelig å planlegge for noe man ikke vet kommer. Likevel beskriver respondentene hvordan man generelt sett overlater endringsønsker til ledelsen og man får inntrykk av at de uforutsette hendelsene blir håndtert på en grei måte når de oppstår. Når jeg stiller spørsmålet; hva med uforutsette hendelser? Svarer Rådgiver på en noe alternativ måte: *”Her? Da løper de for å fikse! Umiddelbart.. Disruptive..”* (Rådgiver).

Ettersom avdelingen beskriver seg selv som smidige, og til dels som scrum-brukere hadde jeg forventet en utstrakt bruk av delleranser. Dette viste seg imidlertid ikke å stemme da tre av fire i ledelsen beskriver at de ofte leverer prosjekter i én leveranse på slutten. Dette forklares i stor grad med størrelse på prosjektene og at det er vanskelig å dele små oppdrag inn i flere leveranser. Dette kan også forklares av mangelen på særlig bruk av sprinter hvor det ville vært naturlig med en ferdig funksjonalitet på slutten av hver sprint. Også utviklerne som er utleid og jobber på et prosjekt med mye større bruk av Scrum beskriver hvordan de kunne vært flinkere med delleranser og ferdige deler i forbindelse med sprint review.

”Vi kjører, vi er ikke så flinke med demoene, at den sprint reviewen kan du si, der er vel det punktet i scrum-metoden som ikke vi er flinke på. Altså, at vi viser av og til litt av det vi har gjort...” (Utleid/Back 2)

At selskapet som helhet var ISO-sertifisert var ikke noe overraskelse og med tanke på størrelsen på konsernet var dette ventet. Det jeg derimot fant interessant var det faktum at ikke mange respondenter var klar over denne sertifiseringen og at de ikke virket bekymret over at de burde være det. En av utviklerne beskriver det slik:

”Men, og sånn er det gjerne med litt sånne store system [ISO], tregheten er åpenbar. Det tar tid å komme inn i rutinene og det tar tid før man vet at det finnes til og med. Fordi ingen har fortalt oss det enda” (Frontend 2).

Dette er et utsagn som tydelig er påvirket av det faktum at avdelingen nylig ble kjøpt opp og at man dermed har en ny øvre ledelse, organisasjon, kontorer og en rekke andre ting å forholde seg til. Også en av avdelingslederne beskriver at han regner med de må adoptere ISO etter hvert.

Leder beskriver et noe annet synspunkt på denne type sertifisering. Han forklarer at man er nødt for å ha dette i konsernet på grunn av de andre, større avdelingene som opererer med tunge systemer med høyere risiko, men at man i denne avdelingen har mer en holdning om at: *”Så er det mer opp til oss her å se ok, hva er risikoen, hvor mye risiko kan vi ta. Hva er kunden villig til å betale for?”* (Leder).

Jeg valgte å spørre respondentene om hva de mente som gjorde bedriften smidig og fikk en rekke ulike svar som beskrevet i kapittel 5.2.7. Det som også var interessant med dette var de

ulike respondentenes tolkning og forståelse av begrepet ”smidig”. En av respondentene var ikke engang klar på hva det innebar og svarte: *”Sånn når det gjelder utviklingsmetoder og metodikk å sånn, det kan jeg ingenting om selv om jeg sikkert bruker det. Så jeg vet ikke helt hva du mener med det..”* (Backend 1).

Likevel var de fleste klare på betydningen av begrepet smidig og noen av respondentene var også mer bevisste på bruken og forståelsen av begrepet:

”... smidig er et dumt begrep når folk ikke kjenner tanken bak det. Smidig er jo et vidt begrep. Men når de fleste sier smidig så mener de jo Scrum. Og smidig er noe som er strikt, og sånn som XP, men skal man gå på manifestet, så kan man jo nesten gjøre som man vil, så det kan jo nesten bli en ametodisk måte å jobbe på” (Utleid/Back 1).

Et annet interessant spørsmål å stille var hvem respondentene mente la føringene for rutinebruk. Her svarte de både kunden, ledelse, bedriften og utvikleren. At jeg fikk så ulike svar på dette spørsmålet var ikke ventet. Likevel var det som forventet at ledelsen stort sett ville svare at det var de eller bedriften som la føringene. At en av utviklerne var tydelig på at det var han som utvikler som la føringene var et interessant utsagn, spesielt sett opp mot at de andre utviklerne som svarte på spørsmålet ga svaret kunden. En mulig forklaring på dette kan være at utvikleren som svarte slik er også den utvikleren med lengst erfaring i avdelingen.

For å oppsummere kan det se ut som rutine for smidig systemutvikling består av:

1. Kunde tar kontakt med ledelse for å gi oppdrag.
2. Prosjektleder eller avdelingsleder gjør estimater og gir et prosjektforslag med pris og tid.
3. Man avholder et møte med kunden for å sette opp spesifikasjoner. (I noen tilfeller gjøres et forprosjekt.)
4. Prosjektleder deler inn i mindre oppgaver og fordeler disse blant utviklerne.
5. Frontendutviklerne planlegger design og oppsett av systemet.
6. Statusmøter avholdes ukentlig, men da ikke per prosjekt, men et større møte som omfatter alle prosjektene som alle utviklerne jobber på.
7. Eventuelle endringer tas i mot av både utviklere og prosjektleder, men det er meningen at det alltid skal videresendes til prosjektleder.
8. Kvalitetssikring blir gjort ut i fra hva kunden er villig til å betale for. Sjelden mer enn testing (som beskrevet i kapittel 5.3.1).
9. Systemet leveres vanligvis som komplett løsning i én levering til kunden.

5.3.3 Kontekstuelle elementer som påvirker rutinene

Utviklerne og deres dynamikk kom tydelig frem som en forklarende faktor da disse kjenner hverandre godt og i de fleste tilfeller har de det klart for seg hvem som fungerer best i hvilke situasjoner. De vet hvordan å fordele arbeid i prosjektene og gjør ofte mye av de samme løsningene fra prosjekt til prosjekt. Med gjenbruk av kode og løsninger kan dette se ut til å føre til at det ikke er nødvendig for utviklerne og gjenoppfinne hjulet for hvert prosjekt og jeg fikk blant annet høre fra en av utviklerne at: *”Noen ting ligner jo mye på ting man har gjort før, så da kan man bruke en del av det”* (Frontend 1).

Når arbeidet oppfattes som relativt enkelt kan det også se ut som at utviklerne kan gjøre omtrent det samme i hvert av prosjektene (på tross av prosjektenes ulike størrelser og kompleksitet) og det er dette som fremtrer som rutinene de bruker, selv om de ikke bevisst identifiserer det som rutiner.

Et annet interessant element ved min undersøkelse er det faktum at oppkjøpet fremdeles er svært nytt og når jeg snakket med de ansatte i mars 2013 kommer det klart frem at de ennå er i innkjøringsfasen og at mye fremdeles oppfattes som nytt av de ansatte. De felles systemene som det nye konsernet krever at alle de ansatte bruker oppleves enda som forvirrende og det brukes tid på å omstille seg til den nye arbeidshverdagen.

”... og etter at jeg kom inn i januar så har ikke jeg fått satt meg inn i det fordi det er seks forskjellige systemer jeg har måtte trykke under, måttet lære meg. Det er timesystem, det samme har jeg måtte lære med fakturering og vi har reiseregninger og det er et eget system for det, så er det et annet system for faktura som jeg skal godkjenne som er kommet inn og så er det kompetansematriser og system for det og så er det jo intranett ja.. Så det er så mange, ja og så har vi også CRM, så det er så mange ting å sette seg inn i når vi er nye” (Avdelingsleder/Prosjektleder).

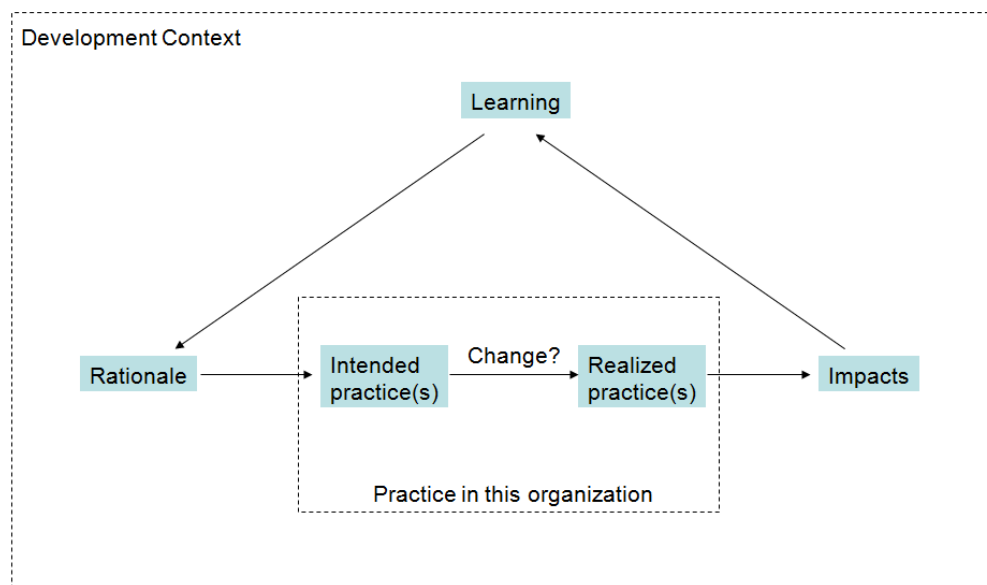
Til slutt ønsker jeg også å påpeke at de synspunktene jeg har fått fra respondentene neppe representerer hele organisasjonen. Det ble nevnt flere ganger under samtalene at dersom jeg hadde intervjuet en annen, større avdeling i stedet, så ville de hatt langt flere klare rutiner og strenge regler for hvordan de jobbet, og med mer overhead fra ledelsen. Forklaringen på hvorfor det ville vært slik i andre avdelinger, i motsetning til de jeg snakket med, er at man der jobber med langt større og mer kritiske systemer, som for eksempel bankløsninger etc

”... hadde du reist til for eksempel Oslo-miljøet, så tror jeg du ville funnet ut at de gjennomfører det [faste rutiner for testing] slik som de skal. De sitter på prosjekter som er på mange tusen timer, og da kan man allokere ressurser på den måten ” (Backend 2).

6 Diskusjon

I dette kapitlet vil jeg forsøke å sette resultatene jeg fant fra min forskning opp mot litteraturen. Hovedfokuset for diskusjonen vil være å se funnene opp mot rammeverket fra Figur 1, gjengitt her som Figur 5. I tillegg vil jeg forsøke å se funnene opp mot problemstillingen samt beskrive bidrag til praksis, se på teoretiske implikasjoner samt eventuelle svakheter i datamaterialet.

6.1 Funn sett opp mot rammeverket



Figur 5 - A framework for describing ISD practice in context (Päivärinta et al., 2011)

Jeg vil videre ta for meg de mest relevante prinsippene fra dette rammeverket og diskutere disse opp mot hva jeg fant i mine undersøkelser, da med spesielt fokus på definert og utført rutine samt endring. (I modellen brukes henholdsvis begrepene ”Intended practice” og ”Realized practice” for det jeg kaller definert og utført.) Læring vil også være et spesielt interessant tema for diskusjonen.

Hva som ligger i begrepet *definert rutine*, kan tolkes på flere måter. Det kan være snakk om hvordan den er definert av designeren eller planleggeren av rutinen eller hvordan den er ble nedskrevet, som ikke nødvendigvis er likt hva designeren hadde planlagt. I tillegg kan det være interessant å skille mellom hvordan rutinen blir utført og hvordan de som utfører den, faktisk beskriver rutinen, da ikke alle har like god selvinnsikt og evne til å beskrive.

Som nevnt tidligere hadde jeg en antagelse om at dette caset ville ha en rekke forhåndsdefinerte og planlagte rutiner som skulle brukes av de ansatte. Det første jeg ønsker å

skille mellom da er *hvem* som har definert rutinene. Det vil si om rutinen er definert av avdelingen og lederen for denne eller av konsernledelsen i større skala. Hvis man ser på organisasjonen som helhet, altså hva hovedkontoret og konsernet mener, så har de tydelig mange definerte rutiner og planer for hvordan ting egentlig skal gjøres, som for eksempel bruk av ISO som i utgangspunktet dikterer hvordan man skal gjøre kvalitetssikring (QA).

”... [organisasjonen] ble resertifisert på ISO nå 01.02.13 tror jeg, så det ligger jo i hele selskapet i forhold til clarity og QA, så ligger det en sånn ISO-sertifisering. Og det må man ha i og med at vi drifter i disse bankløsningsene... Så det er veldig mye av det som er bygd opp i selskapet, er jo bygget rundt de store kundene og de store prosjektene” (Leder).

I tillegg til ISO nevnes det at man har klare testplaner som finnes i organisasjonen og som dermed kan beskrives som en definert rutine for test av konsernledelsen. Ser man på de utførte rutinene som beskrevet av respondentene, opp mot organisasjonens definerte rutiner får man dermed en stor forandring i form av at de ansatte jeg har snakket med verken bruker testplanene eller ISO i sine prosjekter.

Ser man derimot på de definerte rutinene som definert og avklart av *avdelingen* og lederen for denne, får man et noe annet syn på hvordan rutinene er planlagt. De er riktignok ikke planlagt i den forstand at de er nedskrevet og strengt håndfaste, men det fremgår av intervjuene at det finnes en arbeidsmåte og Leder ytrer ingen uenighet i hvordan den gjennomføres av de ansatte. Dette peker da i retning av at rutinene gjennomgår lite endring. Leder beskriver også hvordan man bevisst unngår å belemre utviklerne med konsernets retningslinjer som man ser fra sitatet i kapittel 5.3.1 og kan tolke dette til at man ønsker å bruke avdelingens definerte rutiner som alternativ som alternativ til ”konsernets retningslinjer”.

Respondentene i avdelingen viser ikke et bevisst forhold til arbeidsrutiner samt at det ikke finnes noen klar plan på hvordan disse skal være. I samtale med de ansatte i avdelingen kommer det frem at man likevel gjør arbeid på en ”rutinelignende” måte. Ved sammenligning av utsagnene kan man se visse likhetstrekk i hvordan de ansatte beskriver arbeidet sitt og at det er noen trekk som går igjen i fra prosjekt til prosjekt. Dette fremtrer da som den utførte rutinen. Hva de utførte rutinene innebærer er beskrevet i analysen, men det som er interessant i forbindelse med utført rutine er hvorvidt det er noen forskjeller mellom hva som er definert og hva de ansatte faktisk gjør.

Ser man de utførte rutinene til avdelingen opp mot hva organisasjonen som helhet og øvre ledelse har tenkt for virksomheten så er det en klar endring i hvordan de ansatte gjør arbeidet sitt. Der konsernet er ISO-sertifisert har flere av respondentene ikke hørt om dette og bruker det heller ikke i sin hverdag. Avdelingen velger å se bort fra disse rutinene og gjør heller arbeidet slik de er vant med fra før oppkjøpet, noe som også Lederen ser ut til å ikke bare akseptere, men også oppfordre. Sett ut fra rammeverket så skal en bedrift som i utgangspunktet er ISO-sertifisert ha klare dokumenterte rutiner som blir fulgt i praksis, hvor da tenkt og gjennomført rutine er like. I realiteten så ser ikke avdelingen ut til å samsvare med rammeverket ettersom ISO-sertifiseringen ikke ser ut til å ha noe videre påvirkning på arbeidet i avdelingen.

På spørsmål om respondentene har lagt merke til særlig endring av rutiner selv, enten etter oppkjøpet eller sett over lengre tidsperioder, får man svar om at det ikke har vært noen endringer. De beskriver at de arbeider godt sammen og det fremgår en oppfattelse om at det ikke er nødvendig å endre noe. Man kan da spørre seg om de velger å beholde sine rutiner og arbeidsmåter fordi det faktisk fungerer svært godt, eller om det har andre årsaker som en

motvilje til endring, dårlig evne til å lære av erfaring eller generell uvitenhet om hvorvidt rutinene kunne vært bedre eller ei.

Rutiner i en organisasjon beskrives i litteraturen som en kilde til endring, fleksibilitet og forbedring (Feldman, 2000; Feldman & Pentland, 2003). Basert på dette kan det se ut som at avdelingen, ved å beholde sine systemutviklingsrutiner, går glipp av muligheter til forbedring og enda mer fleksibilitet som igjen samsvarer med avdelingens ønske om å være smidige. Det som av flere respondenter beskrives som smidig i bedriften (se kapittel 5.2.7) er evnen de har til å tilpasse seg endringer i prosjekter raskt. De kan påta seg ulike oppdrag fra forskjellige kunder og definerer seg selv som smidige på bakgrunn av at de kan håndtere disse forandringene og likevel levere. På tross av dette kan man argumentere for at de går glipp av forbedringspotensial i sin arbeidsmåte fordi de fortsatt ikke endrer rutinene for å håndtere de prosjektrelaterte endringene. På den andre siden kan det oppfattes som positivt at det ikke er nødvendig å endre rutiner for å tilpasse arbeidsmåten til ulike prosjekter. Dette er fordi rutinene tyder på at de er så pass gode, at de er i stand til å håndtere de ulike prosjektene uten endring.

Lyytinen og Robey (1999) samt Espedal (2006) beskriver i tråd med rammeverket at man lærer av erfaring. Videre beskriver de hvordan endring er nødvendig for å unngå feil og problemer.

“To avoid learning failure, it is necessary for IS developers to learn from their own and others’ experience and to use this knowledge to change their development practices” (Lyytinen & Robey, 1999, p. 86).

Det er én av respondentene som derimot viser et klart ønske om endring og denne avdelingslederen jobber med å innføre et system for prosjekttrom som nærmere beskrevet i kapittel 5.2.3. I tillegg er dette en av respondentene som tydelig svarer at han regner med at de blir nødt til å forholde seg mer til øvre ledelse og bestemmelser som ISO og lignende etter hvert. Innføringen av prosjekttrom kan bli en utfordrende oppgave og det beskrives i litteraturen hvordan endring er vanskelig og man kan herfra trekke paralleller til såkalt Software Process Improvement eller SPI.

SPI beskrives som utfordrende i litteraturen på grunn av elementer som organisasjonspolitikk, kynisme basert på tidligere mislykkede endringsforsøk og en følelse av at SPI kommer på bekostning av skikkelig arbeid (Herbsleb et al., 1997).

Jeg har ikke fått noe inntrykk fra respondentene at man har forsøkt å drive noe større SPI-prosjekt, men kynismen til endring kommer likevel frem i utsagn som *“... det skorter jo ikke av forslag og tiltak. Men det er gjennomføringen av det som er vanskelig”* (Utleid/Back 1). Det blir i tillegg beskrevet av en av respondentene at det er en avveining om man ønsker å stoppe ordinært arbeid for å kunne innføre endringer på en ordentlig måte.

Skepsis til forandring er også noe som går igjen i litteraturen da man gjerne snakker om motstand og motvilje (Ngwenyama & Nørbjerg, 2010; Pries-Heje et al., 2008). Mennesker er vanedyr og vi liker ikke alltid forandring. Likevel er forandring noe man som konsulent må kunne håndtere.

“Det er en del av greia, det er en del av det som er å være konsulent. At du er, at man må omstille seg hele veien. Man kan ikke ha et fast mønster ” (Utleid/Back 2).

Effekten av rutinene som brukes er god i den forstand at bedriften gjør det greit, de tjener penger, de går med overskudd. Likevel ser det ikke ut som de har noe bevisst forhold til

læring av disse effektene og dette kommer spesielt tydelig frem i mangelen av evaluering. De få respondentene som uttaler seg om evaluering sier alle at de kunne vært bedre på dette. Med mangel av dette kan man argumentere med at de i stor grad ikke lærer noe. Individuelt kan man argumentere for at noen av de ansatte lærer noe i den forstand at de ønsker å endre visse ting, men i de fleste tilfeller blir dette nedstemt fordi det koster for mye penger å ta seg tid til virkelig å se på denne lærdommen, og bruke den til noe fornuftig i form av reviderte og forbedrede rutiner.

Læring beskrives av Päiväranta et al., (2011) som nødvendig og gir uttrykk for at man gjerne må være i en form for læringsløkke. *"...locates ISD practices in a learning loop that is situated in a development context"* (Päiväranta et al., 2011, p. 620). I akkurat dette caset kan det på tross av dette se ut som at de ikke er i noen læringsløkke. De virker ikke bevisste på evaluering og sporing av effekter og de lærer dermed heller ikke av sine rutinevalg, nettopp fordi det ikke måles eller vurderes hvorvidt rutineene fungerer bra eller dårlig.

Organisasjonslæring beskrives av Wang og Ahmed (2003) som et konsept hvor kontinuerlig forbedring er et poeng. Dette er i samsvar med rammeverket og tanken er at man gjennom organisasjonslæring skal kunne få evnen til å forstå og ivareta interne og eksterne endringer. I avdelingen jeg har undersøkt er det vanskelig å identifisere noe klart forhold til organisasjonslæring. Videre snakker man gjerne om taus kunnskap i forbindelse med dette konseptet. I måten de ulike respondentene beskriver rutineene på, kan man tolke det dit hen at de innehar mye taus kunnskap. Det er flinke folk som vet hva de holder på med og det er ikke nødvendig med mye eksplisitt kunnskap. Leder beskriver sine ansatte på denne måten:

"Så det tror jeg er grunnen til at vi lykkes med det vi gjør her, det er at vi kan kjøre veldig mye selvstendige prosjekter på enkeltmannshånd uten at vi går helt av skaffet på kvalitet og leveranser. Det at vi har dyktige erfarne mennesker. Noen som skjønner kunden" (Leder).

Individuelt kan det likevel tidvis se ut som at noen av de ansatte er mer bevisste på læring, eksempelvis Avdelingsleder/Prosjektleder som ønsker endring i form av prosjekttrom. Men som et team og som ledelse fremgår det at man ikke har tid nok til å "stoppe opp" slik at man kan bevisstgjøres lærdommen og bruke den som begrunnelse for å innføre nye tankeretninger.

I tillegg til respondenten som ønsker endring i form av bedre prosjektstyring får man flere andre ganger inntrykk av at det er et underliggende ønske om endring i bedriften. Det uttrykkes at de "alltid" har for mye å gjøre, de mangler folk og de sliter med å "slukke branner". Alt dette kom frem under observasjon av statusmøte og det kan se ut som det finnes et underliggende ønske om endring av denne situasjonen. Dette er alle ting som tyder på at bedriften kanskje burde vurdert å innføre nye rutiner og lære noe av det de gjør. Likevel virker det ikke som det er noe ledelsen har tenkt på eller som de ønsker å bruke tid på.

Metodebruk, eller rettere sagt mangelen av dette, forklares av flere med det faktum at de har så pass små prosjekter at det ikke er noe poeng. I tillegg ser man at kundene er relativt små bedrifter og de er ikke villige til å betale for en formalisert metode med mengder av test, møter og kvalitetskontroll. De er stort sett villige til å ta risikoen. *"Her er vi jo mye mer sentrert rundt SMB-kundene, det er de som misliker at du bruker 10 000 på noe."* (Rådgiver).

Kunden vil som regel ha produktet så enkelt, raskt og billig som mulig, og så lenge det funker så er det stort sett greit. På tross av at de selv mener at de har en smidig metode og at denne har opphav i Scrum så er det svært lite som minner om noe klar metode og enda mindre som minner om Scrum. Man kan faktisk argumentere for at bedriften kjører en noe mer ametodisk

tilnærming til hvordan de utvikler. Ja, de kan endre seg raskt og liker å jobbe kreativt, men det er nesten det eneste som kan defineres som smidig i denne avdelingen.

”... mye av det er på grunn av prosjektstørrelsene, altså å ha et enmanns scrum-team er tullete. Å ha et tomanns scrum-team er på grensa til tullete.” (Utleid/Back 2)

”Metoden” som beskrives av respondentene kan i beste fall defineres som det Fitzgerald et al., (2002) kaller en method-in-action. Ved å se på rammeverket som presenteres av forfatterne (Figur 3) ser man at utviklingskonteksten i stor grad former den metoden man faktisk ender opp med å bruke. Det ser ut til å være spesielt sant i avdelingen da det er vanskelig å kjøre en formalisert metode som Scrum i situasjoner hvor:

- Man har små prosjekter med bare en én eller to utviklere i arbeid.
- Man har prosjekter som skal kjøres for minst mulig penger og kunden ønsker dermed ikke å betale for omfattende metodebruk med mye møter og lignende.
- Man ikke har mulighet til å ha kunden tilgjengelig hele tiden.
- Utviklerne jobber i prosentandeler på en rekke ulike prosjekter og man kan da ikke isolere utviklerne slik at de kan fokusere på ett scrum-team.

Alle disse punktene er situasjoner som avdelingen er i og typen prosjekter de kjører dikterer dermed i stor grad hvordan de kan bruke metode. Fitzgerald et al., (2002) definerer metode som en sammenhengende og systematisk tilnærming som er basert på en spesifikk filosofi for utvikling. Ved sammenligning av denne definisjonen med hva som gjøres i avdelingen ser man at arbeidsrutinene knapt kan defineres som noen form for metode, formalisert eller annet.

Rutinene jeg har trukket ut fra svarene til respondentene kan på noen områder minne mer om det Truex et al., (2000) beskriver som en ametodisk tilnærming. Forfatterne beskriver ametodisk utvikling som et naturlig utfall av komplekse prosesser og at det er en skapelsesprosess der tilfeldige ting kan skje. Videre beskriver de at man ofte har gjensidig avhengige prosesser samtidig og at utvikling skjer på unikt vis. Individualitet er også et av begrepene som brukes for å beskrive ametodisk tilnærming (Truex et al., 2000). Ved sammenligning av disse trekkene med utsagn fra respondentene ser man for eksempel at utviklerne ofte jobber individuelt og at de velger sin unike måte å arbeide på. Tilfeldighet og kompleksitet i prosjektene beskrives også av flere.

Avdelingen har kanskje tatt utgangspunkt i Scrum, men det er ikke mye som ligner på metoden i hva de gjør til daglig. På tross av mangelen på faste scrum-elementer som bruk av sprint, daily scrum, brukerhistorier og iterative leveranser som beskrevet av Schwaber (2004), kan man fortsatt gjenkjenne en del av de smidige elementene fra det agile manifestet. Individet og evnen til å ta egne beslutninger, samarbeid med kunden og evnen til forandring underveis i prosjektene er elementer fra manifestet som samsvarer med resultatene fra undersøkelsen (Beck et al., 2001).

I artikkelen som beskriver rammeverket ytrer forskerne et ønske om at man i fremtiden kanskje kan få på plass et bibliotek over rutiner som fungerer bra. Dette kan også beskrives som en slags parallell til såkalte ”design patterns” som man gjerne bruker i programmering for å standardisere utformingen på koden. De beskriver at man i lys av rammeverket kan observere og definere eksisterende rutiner og praksis på en måte som gjør at man kan ende opp med denne typen database.

“Yet another research opportunity is to observe and classify existing research and ISD methods in light of the framework and come to conclusions about which parts are really studied and what are their combined conclusions. One possible result of such effort could be a database of practice descriptions and patterns in ISD (Wynekoop and Russo 1997), which would make the theory component explicit and well organized according to the structure defined by our framework.” (Päivärinta et al., 2011, p. 621)

Det jeg har sett gjennom forskningen min er at jeg kan ha funnet en slags testrutine som kunne ha fungert i denne typen database eller bibliotek. Rutinen for smidig systemutvikling er noe mindre håndfast og har veldig mange betingelser for at den skal kunne gjenskapes så jeg ser ikke like på denne som like egnet.

Testrutinen jeg fant i avdelingen beskrives på mange måter ulikt av respondentene, men som jeg skriver i slutten av kapittel 5.3.1 fremtrer det en slags rutinebeskrivelse som er gjenkjennelig likevel. Dette er en rutine som er relativt enkel og som tydeligvis fungerer i kontekst av små prosjekter med relativt lav risiko hvor kunden ikke ønsker å betale så mye. Betingelsene for denne rutinen blir dermed at det er et lite prosjekt med lav kompleksitet. Essensen er at man ikke tester stort mer enn man må for å få et ”godt nok” produkt. Dette beskrives av Rådgiver som at *”man må vite når godt nok er godt nok... Det trenger ikke være perfekt”* (Rådgiver).

Denne forenklete testrutinen som fremtrer, beskrives som at den ikke har endret seg i senere tid. Det viser da at den ses på som tilstrekkelig og at den dermed videreføres som en velfungerende rutine. Effekten av rutinen ble ikke beskrevet i klartekst fra noen av respondentene, men det kan se ut som at den videreføres fordi den er enkel, de ansatte er komfortabel med den og den bruker minimalt med ressurser. Noe som oppleves som positivt både fra de ansatte som ikke må bruke mye tid på rutinen i tillegg til kunden som ikke må betale mye for testing. Basert på dette kan man argumentere for at denne testrutinen kunne vært plassert i en rutinedatabase som beskrevet av Päivärinta et al., (2011).

6.2 Funn sett opp mot problemstillingen

Som nevnt i innledningen har problemstillingen gjennomgått en viss utvikling basert på dataene jeg samlet inn. I utgangspunktet ønsket jeg å kartlegge forskjellene mellom de definerte og de utførte rutinene og hvorvidt dette førte til bevisst endring og gjerne innføring av nye eller reviderte rutiner.

I begynnelsen av min forskning, før jeg gjorde intervjuer, antok jeg at dermed at avdelingen i mye større grad ville følge rammeverket i Figur 5 og at man ville ha strenge krav fra konsernledelsen i form av faste definerte rutiner som var nøye planlagt og i tråd med konsernets visjon og ønsker. Jeg hadde også en antagelse om at man ville ha en relativt stor grad av endring, både i form av definert vs. gjennomført rutine, men også en generell utvikling av rutinene basert på effekt og lærdom. Det som derimot kom frem var at avdelingen ikke brukte mange av konsernets rutiner og at de i hadde en stor grad av selvbestemmelse. Videre kom det frem at de heller ikke endret rutinene sine bevisst i noen nevneverdig grad.

Spørsmålet jeg da stilte meg selv, ble heller hvorfor velger denne avdelingen å videreføre rutinene? Hva er motivasjonen som gjør at de ikke ønsker eller trenger å endre seg? Hva er forutsetningene for denne videreføringen av rutiner? Problemstillingen jeg derfor endte på,

lyder som følger: *Hvilken motivasjon og forutsetning har man for å videreføre systemutviklingsrutiner i en bedrift?*

For å kunne svare på denne problemstillingen har jeg forsøkt å avdekke hva som ligger i begrepet **motivasjon**. Jeg beskrev i innledningen at dette er det som *forårsaker handling* og aktivitet for mennesker. Faktorene som *styrer mål og mening* og som gir oss *engasjement*. Videre beskrev jeg **forutsetning** som noe som er *påkrevd for at noe annet skal skje*. Hva er *betingelsen* for at en gitt hendelse skal finne sted?

Motivasjon for videreføringen av rutiner fremstår for meg som ganske enkelt at det er lettest. Mangelen på bevisst handling for å endre rutinene vil kreve mindre tid og energi enn å drive med kontinuerlig evaluering og forbedring av rutiner og arbeidsmåter. Mangelen av handling på dette området gjør at avdelingen kan fortsette å fokusere på prosjektene, kundene og hvordan skape best mulig løsninger med minst mulig ressurser. På denne måten får man som Herbsleb (1997) nevner, muligheten til å unngå følelsen av at man kaster bort tid på endring og forbedring når man kunne gjort "skikkelig arbeid".

En annen motivasjon for videreføringen av rutinene kan være på grunn av utsagnet: *"Sånn som vi alltid har gjort"* (Rådgiver). Dette er ikke et uttrykk brukt spesifikt av flere respondenter, men det kommer frem mellom linjene at utviklerne gjør rutinene sine på en gitt måte fordi de ikke husker noe annet. De har drevet med testing på samme måte så lenge de har jobbet i avdelingen og i bedriften før oppkjøpet. Det slår meg at de ikke ønsker å røske opp i noe som de ikke ser på som "i behov av endring".

Motivasjon og forutsetninger henger nøye sammen og det fremstår for meg som at den viktigste forutsetningen, som også er en motiverende faktor, er det faktum at rutinene ser ut til å fungere. Det virker som om de kan bruke rutinene slik de er, uten endring, og fremdeles oppnå gode resultater. Kundene er fornøyde og konsernet var tydeligvis fornøyde nok med resultatene fra den oppkjøpte bedriften til at de ønsket å kjøpe den. Rådgiver beskriver denne årsaken til oppkjøp slik:

"Det er jo derfor de kjøpte oss da. Det er fordi vi, og det er jo også kjennetegnet på en liten organisasjon med kloke hoder, vi er veldig selvdrevne, vanskelig å presse inn i et sånt rutineprega miljø" (Rådgiver).

I en situasjon som dette, hvor ting fungerer, er det liten grunn til å endre på det. I tillegg, som man ser fra sitatet, er det vanskelig å presse denne typen organisasjon inn i faste strenge virksomhetsomspennende rutiner. Dersom de skulle ønske å bruke tid og ressurser på å innføre nye, eller bytte ut rutiner må de kunne ha en forventning om at konsekvensene (altså impact) av rutinebyttet er tilstrekkelig positivt til å rettferdiggjøre kostnadene ved innføring av nye rutine. Tilsvarende vil de ikke ønske å droppe rutiner hvis de ser ut til å ha positiv effekt.

6.3 Bidrag til praksis, råd og anbefalinger til avdelingen

Evaluering fremstår for meg tydelig som et av områdene avdelingen virkelig kan tjene på å gjøre bedre. Med evaluering av arbeidet og rutinene vil de ha større mulighet for læring og bevisstgjøring. Sistnevnte slår meg som svært viktig når man velger å beholde rutiner. For å kunne rettferdiggjøre dette skikkelig må det være et poeng at man har god innsikt i hva man holder på med og hvorfor man gjør det på den måten man gjør.

En løsning ville nok vært å innføre noe mer rigide arbeidsrutiner generelt, men likevel å la utviklerne beholde muligheten til kreativitet og individualitet. Å bevisst velge for eksempel en ametodisk løsning fremgår som noe overilt og ekstremt. Noe form for kontroll og definerte rutiner er nok positivt for denne avdelingen, men likevel fremstår ikke Scrum som løsningen. Det er blitt beskrevet gang på gang av respondentene at en standard scrum-løsning ikke er ideelt på grunn av omstendighetene. Å kjøre en så rigid metode på svært små team som jobber på flere prosjekt samtidig vil ikke fungere og en såkalt method-in-action vil være bedre.

Det fremstår som et poeng fra konsernet at man ønsker å være smidige og dette tror jeg vil være svært positivt å fortsette med. Spørsmålet er mer i hvilket nivå ønsker man å være smidige. For meg fremstår de generelle prinsippene fra The Agile Manifesto (Beck et al., 2001), uten å blande inn en klar metode, som gode konsepter for avdelingen å fokusere på.

6.4 Svakheter og begrensninger i datamaterialet

Som beskrevet i kapittel 3.4 har man noen begrensninger i valgt forskningsdesign som kan påvirke funnene og konklusjonene man kan trekke. Videre har jeg også sett at man får noen svakheter i datamaterialet i form av tvetydighet i utsagn, ulikhet i hvilken mening man legger i disse og min egen rolle som forsker. I tillegg har man elementer som faren for usannhet blant respondentene samt at de oppfører seg og svarer annerledes fordi de vet at de blir forsket på (Jacobsen, 2005).

Noe som kanskje er litt spesielt i forhold til akkurat min forskning er det faktum at jeg ber om svar på hva de ansatte faktisk gjør i hverdagen, sett opp mot hva ledelsen ønsker at de skal gjøre. Dette kan være problematisk da ansatte gjerne ikke vil fremstå som "vanskelige" eller ulojale og dermed endrer svarene sine. Dette er en risiko jeg har vært nødt for å ta i forskningen og jeg har på tross av dette fått inntrykk av at respondentene har vært ærlige. Tidvis har respondentene vært så ærlige om akkurat hva de synes at jeg har blitt bedt om ikke å sitere dem på spesifikke ting.

Funnene jeg har gjort er ikke nødvendigvis generaliserbare eller gjeldende for andre lignende avdelinger. Ettersom jeg valgte et enkelt casestudie er ikke funnene sammenlignet med annet enn litteraturen. Likevel mener jeg det er interessant forskning som går i dybden på akkurat dette caset og som gir et innblikk i hvordan ting foregår i virkeligheten. Forskningen viser hvordan man gjerne er prisgitt omstendighetene som antall prosjekter man jobber på samtidig, hvor mange ansatte man er for tiden samt hva kunden ønsker. Dette er problemstillinger jeg kan tro mange konsultantselskap har, uten at jeg har noe håndfast bevis for dette.

Et annet spesielt element rundt akkurat dette caset er det faktum at de ble kjøpt opp svært nylig. I tiden hvor jeg intervjuet respondentene var man enda i en omveltningssfas med mye nytt og mange fremmede ting å forholde seg til. Det kunne vært svært interessant og gått tilbake til avdelingen om et års tid for å se hvordan de tilpasser seg til oppkjøpet og hvorvidt flere av de definerte rutinene fra konsernledelsen gjør seg gjeldende. Kanskje er avdelingens valg om å beholde sine rutiner noe som blir problematisk etter hvert og som bli erstattet av mer tydelige krav.

6.5 Teoretiske implikasjoner

Studien har vist meg at læring og selvevaluering i en bedrift er vanskelig. For å kunne ta læring av effektene man får av egne rutiner og arbeidsmåter, er det viktig at man har et bevisst forhold til dette. På tross av at avdelingen i dette caset velger å videreføre sine rutiner så betyr det ikke nødvendigvis at det er fordi de bruker de beste løsningene. Det kan også bety at avdelingen ganske enkelt ikke vet hvordan, man kunne gjort ting bedre, eller at de ikke ser nok insentiv til å gjøre endringene.

Ettersom dette er en enkeltstående studie av en relativt spesifikk situasjon med en nylig oppkjøpt avdeling i et større konsultentselskap, så er sannsynligvis ikke resultatene særlig generaliserbare. Likevel er det et interessant case hvor det har fremkommet mange tankevekkende funn og disse kan kanskje bidra til å gi en bredere forståelse av systemutviklingsrutiner og måten man jobber på. Det som derimot kanskje kan generaliseres til en viss grad er testrutinen som fremtrer i slutten av kapittel 5.3.1. Dersom biblioteket av systemutviklingsrutiner og praksiser som beskrives av Päivärinta et al., (2011) skulle bli en realitet så kan kanskje deler av denne forskningen stå som et bidrag der.

Det er forsket mye på endring og endringsledelse og hvilke effekter man kan få av endring. I motsetning til dette ser man at det er gjort lite forskning på de situasjonene hvor man velger å videreføre løsningene man har. Det kunne vært interessant og sett hva forskningen hadde funnet om man hadde sett noe videre på dette. Er det virkelig så enkelt som at man lar være å gjøre endring fordi det er enklest? Fordi det er billigst her og nå? Eller har det underliggende årsaker som at man ikke har nok læring i organisasjonen og at man burde gjennomføre endring.

7 Konklusjon og implikasjoner

Studien er blitt gjort gjennom kvalitativ forskning med et enkelt case studie som utgangspunkt. Jeg gjorde en litteraturstudie i forkant av datainnsamlingen som la grunnlag for min forståelse av emnet og mitt ønske for videre forskning. Gjennom studien gjorde jeg noe observasjon, men hovedfokus for innsamling av data var gjennom semistrukturerte intervjuer. Gjennom disse intervjuene forsøkte jeg å avdekke så mye detaljer som mulig rundt to rutiner i avdelingen jeg studerte. Målet var å gjennom undersøkelsen av disse rutinene kunne svare på spørsmålet:

Hvilken motivasjon og forutsetning har man for å videreføre systemutviklingsrutiner i en bedrift?

Motivasjonen for videreføring av rutiner fremstod hovedsakelig som bekvemmelighet. Da både i form av at det er enkelt å slippe å endre noe, men også i form av at det ”alltid har vært sånn”. Man gror litt fast i gamle rutiner og man får dermed ikke noe motivasjon for faktisk å gjennomføre endringer.

Motivasjon og forutsetninger henger sammen og den viktigste faktoren for videreføring av rutiner er det faktum at de de gjør nå tydeligvis fungerer. At de velger å la være å gjennomføre endringer tyder på at de forventede konsekvensene av endring ikke veier opp for kostnadene og ressursene som er nødvendig for dette.

Jeg innser at dette ikke er en sjokkerende åpenbaring om hvordan systemutvikling fungerer i praksis, men det er tankevekkende hvor lite evaluering og selvbevissthet man har rundt hvilke rutiner man velger å bruke og videreføre. Mitt klare råd til avdelingen vil være å bruke mer tid på bevisstgjøring og evaluering slik at man er klar over hvorfor man jobber som man gjør. Før man faktisk en ønsket effekt, eller er man rett og slett for ”lat” til å finne ut om man burde drive rutineendring eller ei? Man må til nesten spørre seg om Lyytinen og Robey (1999) har rett; *“Not only have many organizations failed to learn, but they have also learned to fail.”* (p. 85).

Generaliserbarheten av studien er nok noe begrenset, men det er interessant å se hvordan testrutinen som fremkom av intervjuene kanskje kan tas i bruk av også andre bedrifter da den fremstår som relativt enkel og ressursbesparende. Päivärinta et al., (2011) beskriver ønsket om en rutinedatabase og testrutinen jeg har avdekket i min forskning kan ses på som et potensielt bidrag til denne.

Studien vil forhåpentligvis også kunne bidra med mer bevisstgjøring rundt temaet og gi en dypere forståelse av rutiner og spesielt rutiner sett opp mot metode og hvordan man kan lære av seg selv og erfaringer. Fremover kunne det vært interessant for akkurat dette caset å gå tilbake om et års tid for å se nærmere på hvordan ting gjøres når oppkjøpet ikke er så nytt. Vil rutinene være like, eller vil de være mer påvirket av konsernledelsen?

Av en mer generell observasjon så kunne det vært interessant å gjennomføre lignende studier i andre konsulentavdelinger for å se hvorvidt man gjennomfører endringer og dersom man velger videreføring; hvorfor? Er det ganske enkelt fordi det krever minst innsats, eller er det fordi man faktisk har vurdert situasjonen og kommet frem til at endring ikke kan rettferdiggjøre kostnadene?

8 Referanser

- Beck, Kent, Beedle, Mike, Bennekum, Arie van, Cockburn, Alistair, Cunningham, Ward, Fowler, Martin, . . . Thomas, Dave. (2001). Manifesto for Agile Software Development. Retrieved 23.05, 2013, from <http://agilemanifesto.org/>
- Boehm, Barry W., & Turner, Richard. (2003). *Balancing agility and discipline : a guide for the perplexed*. Boston, Mass.: Addison-Wesley.
- businessdictionary.com. (2013). Organizational Learning. Retrieved 02.06, 2013, from <http://www.businessdictionary.com/definition/organizational-learning.html>
- Bygstad, Bendik, & Munkvold, Bjorn Erik. (2007). *The significance of member validation in qualitative analysis: experiences from a longitudinal case study*. Paper presented at the System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on.
- Cockburn, Alistair. (2007). *Agile software development : the cooperative game*. Upper Saddle River, NJ: Addison-Wesley.
- Cohen, Michael D., & Bacdayan, Paul. (1994). Organizational Routines Are Stored as Procedural Memory: Evidence from a Laboratory Study. *Organization Science*, 5(4), 554-568.
- Dey, Ian. (1993). *Qualitative data analysis: A user-friendly guide for social scientists*: Routledge.
- Dubé, Line, & Robey, Daniel. (1999). Software stories: three cultural perspectives on the organizational practices of software development. *Accounting, Management and Information Technologies*, 9(4), 223-259. doi: 10.1016/S0959-8022(99)00010-7
- Espedal, Bjarne. (2006). Do Organizational Routines Change as Experience Changes? *Journal of Applied Behavioral Science*, 42(4), 468-490. doi: 10.1177/0021886306291601
- Feldman, Martha S. (2000). Organizational Routines as a Source of Continuous Change. *Organization Science*, 11(6), 611-629.
- Feldman, Martha S., & Pentland, Brian T. (2003). Reconceptualizing Organizational Routines as a Source of Flexibility and Change. *Administrative Science Quarterly*, 48(1), 94-118.
- Fitzgerald, Brian, Russo, Nancy L, & Stolterman, Erik. (2002). *Information systems development: Methods in action*: McGraw-Hill London.
- Hansen, Tore Berg, & Hjertø, Greta. (2003). *Kvalitet og programvareutvikling*. [Trondheim]: TISIP.
- Herbsleb, James, Zubrow, David, Goldenson, Dennis, Hayes, Will, & Paulk, Mark. (1997). Software quality and the Capability Maturity Model. *Commun. ACM*, 40(6), 30-40. doi: 10.1145/255656.255692
- Iivari, J., Hirschheim, R., & Klein, H.K. (2001). *Towards more professional information systems development: ISD as knowledge work*. Paper presented at the Proceedings of the 9th European Conference on Information Systems, Bled, Slovenia.
- Jacobsen, Dag Ingvar. (2005). *Hvordan gjennomføre undersøkelser?: innføring i samfunnsvitenskapelig metode*. Kristiansand: Høyskoleforl.
- Kvale, Steinar, Brinkmann, Svend, & Anderssen, Tone Margaret A. Rygge Johan f. (2009). *Det kvalitative forskningsintervju*. Oslo: Gyldendal akademisk.
- Lam, Alice. (1998). Tacit Knowledge, Organisational Learning and Innovation A Societal Perspective: DRUID, Copenhagen Business School, Department of Industrial Economics and Strategy/Aalborg University, Department of Business Studies.

- Larsen, E., Päivärinta, T., & Smolander, K. (2008). *The NIPO Grid - A construct for studying systems development practices in organizations*.
- Lyytinen, Kalle, & Robey, Daniel. (1999). Learning failure in information systems development. *Information Systems Journal*, 9(2), 85-101.
- Myers, Michael D, & Newman, Michael. (2007). The qualitative interview in IS research: Examining the craft. *Information and organization*, 17(1), 2-26.
- Ngwenyama, O., & Nørbjerg, J. (2010). Software process improvement with weak management support: An analysis of the dynamics of intra-organizational alliances in IS change initiatives. *European Journal of Information Systems*, 19(3), 303-319.
- Oates, Briony J. (2006). *Researching information systems and computing*. London: Sage Publications.
- Page, Alan, Johnston, Ken, & Rollison, Bj. (2009). *How We Test Software at Microsoft®*: Microsoft Press.
- Pries-Heje, J., Aaen, J.N.I., & Elisberg, T. (2008). The Road To High Maturity. *Software Processes & Knowledge*, 165 - 195.
- Päivärinta, Tero, Smolander, Kari, & Larsen, Even Åby. (2011). Towards a Framework for Building Theory from ISD Practices. In J. Pokorny, V. Repa, K. Richta, W. Wojtkowski, H. Linger, C. Barry & M. Lang (Eds.), *Information Systems Development* (pp. 611-622): Springer New York.
- Päivärinta, Tero, Smolander, Kari, & Larsen, Even Åby. (2012). A model for analyzing changes in ISD practices. *JITTA*.
- Schwaber, Ken, & Microsoft, Corporation. (2004). *Agile project management with Scrum*. Redmond, Wash.: Microsoft Press.
- Truex, D., Baskerville, R., & Travis, J. (2000). Amethodical systems development: the deferred meaning of systems development methods. *Accounting, Management and Information Technologies*, 10(1), 53-79.
- Turbit, Neville. (2006). Developing a Test Strategy Retrieved 24.05, 2013, from http://www.projectperfect.com.au/info_test_strategy.php
- Walsham, Geoff. (1995). Interpretive case studies in IS research: nature and method. *European Journal of information systems*, 4(2), 74-81.
- Wang, Catherine L, & Ahmed, Pervaiz K. (2003). Organisational learning: a critical review. *Learning Organization, The*, 10(1), 8-17.
- Yin, R.K. (2009). *Case Study Research: Design and Methods* (4th ed.): SAGE Publications.

9 Vedlegg

9.1 Vedlegg 1 - Intervjuguide

Intervjuguide

Hensikt: Kartlegge og forstå hvordan testrutine og rutiner for smidig arbeid i systemutvikling fungerer hos en lokal avdeling i et større konsulentselskap.

Design: Casestudie
Kvalitativt
Semistrukturerte intervjuer

Før selve intervjuet starter vil jeg bruke noen minutter på å forklare hensikten og temaet for intervjuet. Jeg vil si noe om hva jeg forventer og spørre intervjuobjektet om de ønsker anonymitet eller om de har noen spørsmål eller spesifikke ønsker. Videre vil jeg spørre om tillatelse til å bruke lydopptak og eventuelt starte dette.

Problemstilling:

”Hvilken motivasjon og forutsetning har man for å videreføre systemutviklingsrutiner i en bedrift? Med fokus på testrutine samt rutine for smidig utvikling”

Bakgrunnsinformasjon – deltager

Navn:
Stilling:
Ansatt siden:

Dato og tid for intervjuet:
Intervjuets lengde:
Type intervju:
Lokasjon:
OpptakÅpningsspørsmål

Deltager

1. Hva er ditt navn?
2. Hvilken utdanning har du?
3. Hvilken stillingsbeskrivelse har du?
4. Hvor lenge har du vært ansatt i nåværende stilling?
5. Hvilket prosjekt jobber du på?

Prosjekt

1. Hvilken type er prosjektet?
2. Hvor mange jobber på det? (Omfang)
3. Hva slags tidsperspektiv har man?

Hovedtema

På bakgrunn av innledende samtaler med bedriften ønsker jeg å ta for meg to spesifikke rutiner hvor jeg vil forsøke å avdekke så mye detaljer som mulig om hvordan disse gjennomføres. Den første rutinen er testing og hvordan dette gjennomføres i bedriften. Den andre rutinen er hvordan man bruker smidig arbeid i systemutvikling, det vil si hvordan dette gjøres og hvordan det påvirker både prosjektdeltakere og ledelse/kunde.

Testrutine

1. Hvordan tester du?
2. Har dere en spesifikk testplan som følges?
 - a. Hva inneholder i så fall denne?
 - b. Er denne fast for hver gang man tester?
 - c. Hvem har utviklet denne testplanen?
 - d. Følger denne noe form for fast standard? (Eks. IEEE)
 - e. Når ble testplanen sist endret?
 - i. Hvordan ble den endret?
 - ii. Hvem gjorde (bestemte og innførte i planen) i så fall dette?
3. Hva gjorde du spesifikt, sist du var i en testesituasjon?
4. Utføres testing i det nåværende prosjektet på samme måte som i det forrige du deltok i?
 - a. Hvorfor/hvorfor ikke?
 - b. Hvis endringer ble gjort, kan du peke på noe spesifikt som du tror førte til endring?
5. Benyttes faste kriterier for godkjenning av tester?
6. Når tester dere for første gang i et pågående prosjekt?
7. Hvilke type tester utfører dere?
 - a. Hender det at man planlegger tester som ikke gjennomføres?

8. Hvordan loggføres testene som gjennomføres?
 - a. Brukes faste systemer, skjema eller andre måter å dokumentere testingen og testenes resultater?
 - b. Loggfører alle likt?
 - c. Hvem bestemmer hvordan det loggføres?
9. På hvilket stadium involveres kunden i testing?
 - a. I hvor stor grad får kunden delta?
 - b. Er dette bestemt på forhånd eller er det opp til kunden?
10. Hvorfor tester dere slik som dere gjør?
 - a. Er testrutinen lik for hvert prosjekt?
 - b. Har dere "alltid" testet slik som dette?
 - c. Ble det gjort endringer i denne rutinen etter oppkjøp?
 - d. I så fall: Hvorfor?
11. Tester dere på samme måte i alle typer prosjekter?
 - a. Hvis nei, hvordan er det ulikt?
12. Dersom dere finner feil:
 - a. Hvordan kategoriseres disse?
 - b. Hvordan rapporteres disse?
 - c. Er noen mer alvorlige enn andre og i så fall hva skjer da?
13. Hvordan prioriterer dere rettingen av feil?
14. Når fikses feil?

Rutine for smidig arbeid i systemutvikling

1. Hva slags faste møter gjennomføres i forbindelse med smidig systemutvikling?
2. Hva var det siste møtet du deltok på?
 - a. Opplevde du dette som nyttig/positivt?
3. Hvordan gjennomføres fordeling av oppgaver/arbeid?
4. Hvordan håndteres uforutsette hendelser og endringsønsker?
 - a. Hva skjer når ting ikke går etter planen?
5. Hvordan gjennomføres kvalitetssikring av arbeidet?
6. Hva er rutinene for leveranser i prosjektet?
 - a. Varierer disse mye fra prosjekt til prosjekt? Forklar
7. Er rutinene like fra prosjekt til prosjekt?
 - a. Sammenlign det nåværende prosjektet med det forrige, er det forskjeller mellom hvordan rutinene ble utført?
 - i. I så fall, hva var disse forskjellene?
 - ii. Hvorfor tror du det var/ikke var ulikheter?
8. Hvem legger føringer for systemutviklingsrutinene?
 - a. Kunden?
 - b. Bedriften?
 - c. Andre?
9. Ble smidig arbeid vurdert opp mot alternative arbeidsmåter?