

Thomas Nilsen

Smidig utvikling i en mellomstor bedrift på Sørlandet

Masteroppgave i Informasjonssystemer

Universitetet i Agder

Fakultet for Økonomi og Samfunnsfag

2008

Forord

Denne masteroppgaven avslutter det 2-årige Mastergradsstudiet i Informasjonssystemer ved Universitetet i Agder. Gjennom dette studiet har jeg blant annet tilegnet meg kunnskap om hvordan virksomhetsstøttesystemer kan bidra til effektivitet, verdiskapning i organisasjoner, strategivalg, prosjektstyring og teori om systemutviklingsmetodikker. I denne oppgaven har jeg i løpet av vårsemesteret 2008 gått i dybden på en mellomstor bedrift på Sørlandet for å undersøke deres bruk av den smidige systemutviklingsmetodikken Scrum.

Arbeidet med denne oppgaven og min kontakt med denne bedriften har vært svært verdifull i forhold til min forståelse for bruk av systemutviklingsmetodikker i praksis. Jeg har blitt utfordret til å sette teoretisk kunnskap opparbeidet gjennom studiene opp mot den virkeligheten som eksisterer i den undersøkte bedriften.

De fleste fag jeg har vært gjennom på universitetet har lagt opp til at jeg skal samarbeide med andre, dette har gjort meg klar for et arbeidsliv som legger til rette for samarbeid i lag. Denne oppgaven har jeg derimot valgt å løse alene og dette valget ble tatt da jeg ønsker å vise at jeg som individ har tilegnet meg den kunnskapen og de ferdighetene som trengs for å kunne løse en så utfordrende og tidkrevende oppgave som denne masteroppgaven er. Selv om denne oppgaven kun er skrevet av undertegnede, har jeg vært avhengig av et godt samarbeid med den undersøkte bedriften og mine veiledere. Et samarbeid som jeg mener har vært svært vellykket.

Jeg ønsker med dette å rette en stor takk til bedriften som har latt meg få frie tøyler til å forme, undersøke og jobbe med min masteroppgave i samarbeid med dem. Jeg ønsker også å takke mine veiledere professor Tero Päivärinta og førstelektor Even Åby Larsen for alle samtaler tilbakemeldinger jeg har fått nytte av gjennom prosessen. Til slutt vil jeg takke alle andre som har bidratt til inspirasjon og kunnskap underveis.

Takk!

Kristiansand 2. Juni 2008

Thomas Nilsen

Sammendrag

Systemutviklingsmetodikk og spesielt smidige metodikker og teknikker er et populært emne hos praktikere og forskere i systemutviklerbransjen. Det er gjort flere undersøkelser på hvilke metodikker som blir benyttet i praksis, hvordan metodikker kan endre seg med tid og hvilke faktorer som påvirker valget av hvilken metodikk som velges. Flere undersøkelser ser på metodikken som et helhetlig konsept som blir påvirket av flere faktorer. Færre undersøkelser har analysert de faktiske prinsippene en metodikk inneholder for å se på hvilke faktorer som påvirker bruken av disse.

Denne oppgaven har tatt for seg en mellomstor bedrift på Sørlandet og analysert fire systemutviklingsprosjekter i denne bedriften. Bedriften har tatt utgangspunkt i den smidige systemutviklingsmetodikken Scrum og har innført praksiser fra metodikken samt andre prinsipper hentet fra det smidige miljøet. De fire prosjektene har noe forskjellig utgangspunkt for sin bruk av metodikk. Av prosjekter er det presentert en reiselivsportal, et logistikk-system, en pilot på innføring av Scrum i en ekstern bedrift og en statistikkportal. Disse fire prosjektene har alle hatt forskjellige utfordringer noe som også vises i hvordan hvert enkelt prosjekt har tilpasset sine praksiser til sin virkelighet. Jeg prøver med denne oppgaven å belyse hvilke systemutviklingspraksiser som er planlagt benyttet i bedriften, hvilke praksiser som faktisk er benyttet i prosjektene og hva som eventuelt er bakgrunn for en variasjon mellom det planlagte og det faktiske.

Ved å ta en kvalitativ tilnærming til denne problemstillingen, har jeg funnet frem til en rekke faktorer som har påvirket bedriftens bruk av de enkelte praksiser. Flere av disse faktorene påvirker mer en ett prinsipp, men det er også enkelte faktorer som kun har en effekt på enkelte praksiser benyttet i prosjektene.

Det er funnet at samtlige prosjekter benytter seg av Scrum overordnede valg av praksiser i prosjektet, men at det i enkelte tilfeller er gjort såpass store endringer at målet til metodikken har flyttet seg noe bort fra sitt opprinnelige mål. Bakgrunnen for disse endringene kommer fra flere hold. Det er blant annet funnet at kundens egenskaper, begrensinger og leveranser har en stor påvirkning på hvordan bedriftens prosjekter gjennomfører de praksiser som er definert. Det er også funnet at bruken av et felles prosjektstyringsverktøy i bedriften til en viss grad har formet bruken av de metodiske prinsippene.

Flere av funnene gjort i denne oppgaven har blitt bekreftet av tidligere teori, allikevel kan denne oppgaven bidra til å belyse hvorfor lignende bedrifter tar de valg de faktisk gjør. Dybden i denne undersøkelsen og at den undersøker påvirkningsfaktorer på et nivå som beskriver hver enkelt praksis, kan bidra til en nyansering av tidligere overordnede modeller.

Innholdsfortegnelse

| | |
|--|------|
| Forord..... | III |
| Sammendrag | V |
| Innholdsfortegnelse..... | VI |
| Figuroversikt..... | VIII |
| Tabelloversikt | VIII |
| 1 Innledning | 1 |
| 1.1 Motivasjon for oppgaven..... | 1 |
| 1.2 Problemstilling..... | 1 |
| 1.3 Oppgavestruktur..... | 2 |
| 2 Teoretisk bakgrunn for oppgaven..... | 3 |
| 2.1 Begrepsdefinisjoner | 3 |
| 2.2 Bruk og tilpasning av metodikk..... | 4 |
| 2.3 Smidig systemutvikling | 6 |
| 2.3.1 Smidige metodiske prinsipper..... | 6 |
| 2.4 Scrum | 8 |
| 2.4.1 Griser og kyllinger | 9 |
| 2.4.2 Scrum-team | 9 |
| 2.4.3 Produktbaklogg | 10 |
| 2.4.4 Sprintplanleggingsmøte | 10 |
| 2.4.5 Sprint..... | 10 |
| 2.4.6 Daglige møter..... | 11 |
| 2.4.7 Sprintgjennomgangsmøte | 11 |
| 3 Metodebeskrivelse | 12 |
| 3.1 Forskningsdesign | 12 |
| 3.2 Datainnsamling | 12 |
| 3.2.1 Intervju..... | 12 |
| 3.2.2 Observasjon..... | 14 |
| 3.2.3 Gjennomgang av prosjektdokumentasjon..... | 14 |
| 3.2.4 Litteraturgjennomgang..... | 14 |
| 3.3 Dataanalyse | 15 |
| 3.3.1 Kategorisere data | 15 |
| 3.3.2 Tilordne enheter til kategorier | 15 |
| 3.3.3 Finne sammenhenger i data | 15 |
| 4 Resultater | 17 |
| 4.1 Beskrivelse av organisasjon..... | 17 |
| 4.2 Beskrivelse av prosjekter | 17 |
| 4.2.1 Prosjekt 1 – Reiselivsportal | 17 |
| 4.2.2 Prosjekt 2 – Prosjektstyring hos teknologibedrift..... | 22 |
| 4.2.3 Prosjekt 3 – Logistikkssystem | 24 |
| 4.2.4 Prosjekt 4 – Statistikkportal | 26 |
| 4.3 Analyse av funn i prosjektene..... | 27 |
| 4.3.1 Produktbaklogg | 28 |
| 4.3.2 Scrum-team | 30 |
| 4.3.3 Sprintplanleggingsmøte | 31 |
| 4.3.4 Sprint..... | 32 |

| | | |
|--------|--|----|
| 4.3.5 | Daglige møter..... | 32 |
| 4.3.6 | Sprintgjennomgang..... | 33 |
| 4.3.7 | Brukerhistorier..... | 34 |
| 4.3.8 | Prosjektstyringsverktøy..... | 35 |
| 4.3.9 | Poengestimering..... | 35 |
| 4.3.10 | Kodestandardisering..... | 35 |
| 5 | Diskusjon..... | 36 |
| 5.1 | Faktorer som påvirker bruk av de metodiske prinsippene i organisasjonen..... | 36 |
| 5.1.1 | Faktorens påvirkning i forhold til Scrumprinsippene..... | 36 |
| 5.1.2 | Faktorens påvirkning i forhold til andre prinsipper..... | 39 |
| 5.2 | Hvordan relaterer organisasjonens bruk av Scrum seg i forhold til Scrums idealer?..... | 40 |
| 5.3 | Skiller Scrum seg fra smidig metodebruk generelt?..... | 41 |
| 5.4 | Bidrag til praksis..... | 42 |
| 5.5 | Implikasjoner for forskning..... | 43 |
| 5.6 | Begrensinger i forhold til forskningsmetodikk..... | 43 |
| 6 | Konklusjon..... | 45 |
| | Referanser..... | 47 |
| | Vedlegg..... | 49 |

Figuroversikt

| | |
|---|----|
| Figur 1: Rammeverk for metodebruk (Fitzgerald et al., 2002)..... | 4 |
| Figur 2: Burn-down eksempel fra prosjekt 1 | 7 |
| Figur 3: Scrumprosessen (Schwaber & Beedle, 2002, s8 (gjenprodusert))..... | 8 |
| Figur 4: Forskningsmodell | 12 |
| Figur 5: Veiledning til det kvalitative forskningsintervju (Myers & Newman, 2007) | 13 |
| Figur 6: Dataanalyse modell | 15 |
| Figur 7: Faktorer som påvirker bruk av scrumprinsippet produktbaklogg | 29 |
| Figur 8: Faktorer som påvirker lokalisering av Scrum-teamet | 30 |
| Figur 9: Faktorer som påvirker prinsippet sprintplanleggingsmøtet..... | 31 |
| Figur 10: Faktorer som påvirker bruk av scrumpraksissen sprint..... | 32 |
| Figur 11: Faktorer som påvirker bruk av scrumprinsippet daglige møter | 33 |
| Figur 12: Faktorer som påvirker bruk av scrumprinsippet sprintgjennomgang | 34 |

Tabelloversikt

| | |
|---|----|
| Tabell 1 - Begrepsdefinisjoner..... | 3 |
| Tabell 2 - Generisk NIPO rammeverk | 5 |
| Tabell 3 - Roller i et scrumprosjekt | 9 |
| Tabell 4 - Oversikt over datainnsamling i prosjektene | 12 |
| Tabell 5 - Prosjekt 1: Prosjektdeltakere | 17 |
| Tabell 6 - Prosjekt 2: Prosjektdeltakere | 23 |
| Tabell 7 - Prosjekt 3: Prosjektdeltakere | 24 |
| Tabell 8 - Prosjekt 4: Prosjektdeltakere | 26 |
| Tabell 9 - NIPO grid, organisasjonen | 28 |

1 Innledning

Smidig systemutvikling har tidligere blitt presentert som en revolusjonerende måte å utvikle programvare på. Tidligere var det gjerne personer med spesiell interesse for emnet som implementerte smidige prosesser, men nå har det generelle fagmiljøet i Norge begynt å ta i bruk smidig systemutvikling i større og større grad. Interessen for en mer effektiv systemutvikling er stor, og på Java Brukerforening i Norge sine hjemmesider har de siste 3 av 5 foredrag på deres gruppemøter omhandlet smidige metodiske prinsipper eller verktøy (JavaBin, 2008). Flere har derimot påpekt den faktiske bruken av metodikken i de færreste tilfeller er lik den metodebruken som er planlagt og tilpasninger gjøres til de fleste metodikker og påvist hva som påvirker den faktiske bruken av metodikken (Fitzgerald, Russo, & Stolterman, 2002). En av forkjemperne for bruk av smidige systemutviklingsmetodikker Alistair Cockburn har gjennom sitt arbeide beskrevet flere retningslinjer for hvordan en skal kunne velge den rette komposisjonen av metodiske prinsipper til et prosjekt med visse kriterier (Cockburn, 2000, 2002). Det finnes også eksempler der en kombinerer smidige metodikker for å ta i bruk gode prinsipper fra hver av dem (Fitzgerald, Hartnett, & Conboy, 2006). Samtidig påpekes det gjerne fra metodeforfatterens side at det frafrådes å endre for mye på deres metodikk (Schwaber & Beedle, 2002). Denne retningen mot å se på systemutviklingsmetodikker som enkelte praksiser istedenfor en helhetlig metode fører til at en kan se et mer nyansert bilde av hvordan bedrifter faktisk arbeider i sine prosjekter. Larsen, Päiväranta og Smolander (2007) har kommet frem til rammeverket NIPO grid som kan bidra til, å på en oversiktlig måte, vise den planlagte og den faktiske bruken av metodiske prinsipper i en organisasjon.

1.1 *Motivasjon for oppgaven*

Gjennom Universitetet i Ager har jeg som student blitt introdusert for flere bedrifter innen fagområdet systemutvikling på Sørlandet, og ved en av disse anledningene kom jeg i kontakt med en representant for den undersøkte bedriften i denne oppgaven. Bedriften hadde valgt å benytte seg av Scrum som utviklingsmetodikk, og viste interesse for at jeg kunne undersøke deres bruk av denne metodikken nærmere. Representanten for den undersøkte bedriften hadde i disse samtalene også nevnt at deres bruk av Scrum varierte noe, og at enkelte prosjekter hadde hentet inspirasjon fra flere metodikker definert som smidige, og slik holdt på med å bygge opp en ”verktøykasse” med metodiske prinsipper som kunne benyttes i nåværende og fremtidige prosjekter. Samtidig med dette hadde studiene ved UiA gjort meg interessert i smidig systemutvikling. Jeg fant det spesielt interessant at selv om bedriften hadde nevnt bruk av Scrum, hadde de også benyttet seg av andre metodiske prinsipper og tilpasset prinsippene til hvert enkelt prosjekt.

1.2 *Problemstilling*

Med bakgrunn i at tidligere forskning i større grad fokuserer på hvilke faktorer som påvirker den helhetlige metodikken og kunnskapen om at de fleste gjør tilpasninger til den metodiske bruken i prosjekter har jeg vinklet denne oppgaven til å analysere de enkelte praksiser som blir benyttet i prosjektene i organisasjonen. Problemstillingen jeg har jobbet ut i fra i denne oppgaven lyder som følger:

Hva slags systemutviklingspraksiser blir planlagt benyttet og faktisk benyttet i bedriftens systemutviklingsprosjekter som har tatt utgangspunkt i metodikken Scrum? Hva påvirker valgene, og hva er bakgrunnen for en eventuell variasjon mellom hva som er planlagt og hva som blir utført?

Jeg har også forklart hva som skiller bedriftens praksiser og de praksiser som er beskrevet av Scrumteorien samt gjort en sammenligning av Scrum i forhold til hva som klassifiserer smidig utvikling.

1.3 Oppgavestruktur

Oppgaven er strukturert rundt 6 kapitler.

Kapitel 1: Innledning

Kapitlet plasserer oppgaven i forhold til konteksten, jeg beskriver problemstillingen og, med dette underkapitlet, oppgavens struktur.

Kapitel 2: Teoretisk bakgrunn for oppgaven

Kapitlet består av den teoretiske bakgrunnen for oppgaven. I dette kapitlet er smidig utvikling med fokus på den smidige utviklingsmetodikken Scrum forklart, samt generell forskning på metodiske prinsipper og hvilke faktorer som spiller inn på metodetilpasning i organisasjoner. I dette kapitlet har jeg også presentert begrepsdefinisjoner på begreper som benyttes i oppgaven.

Kapitel 3: Metodebeskrivelse

Kapitel 3 forklarer min metodiske tilnærming til forskningsprosessen. Jeg presenterer datainnsamlingsmetoder, omfang av disse og hvordan jeg har gått frem for å analysere datagrunnlaget som er innsamlet.

Kapitel 4: Resultater

Kapitel 4 beskriver først de prosjekter som er omfattet av denne oppgaven og kjerneinformasjon om hvert enkelt prosjekt. Deretter har jeg sammenlignet de undersøkte prosjektene og analysert funnene gjort. Jeg viser et samlet sett med faktorer som påvirker bedriftenes bruk og tilpasning av de metodiske prinsipper benyttet i organisasjonen.

Kapitel 5: Diskusjon

I kapittel 5 har jeg tatt utgangspunkt i problemstillingen og diskuterer funnene presentert i kapittel 4 mot problemstillingen og det litterære grunnlaget funnet i kapittel 2. Jeg prøver å drøfte de faktorer som er funnet som påvirker Scrum-metodikken og andre metodiske prinsipper funnet i prosjektene. I dette kapitlet har jeg også sammenlignet organisasjonens bruk av Scrum, i forhold til hva Scrum har definert som idealer. Kapitlet avsluttes med at jeg presenterer mitt bidrag til forskning og praktikere i bransjen samt forklarer de begrensninger forskningsmetodikken og datainnsamlingen har i forhold til dette.

Kapitel 6: Konklusjon

Dette kapitlet avslutter oppgaven ved å summere opp de viktigste funnene og mitt bidrag til basen av kunnskap.

2 Teoretisk bakgrunn for oppgaven

2.1 Begrepsdefinisjoner

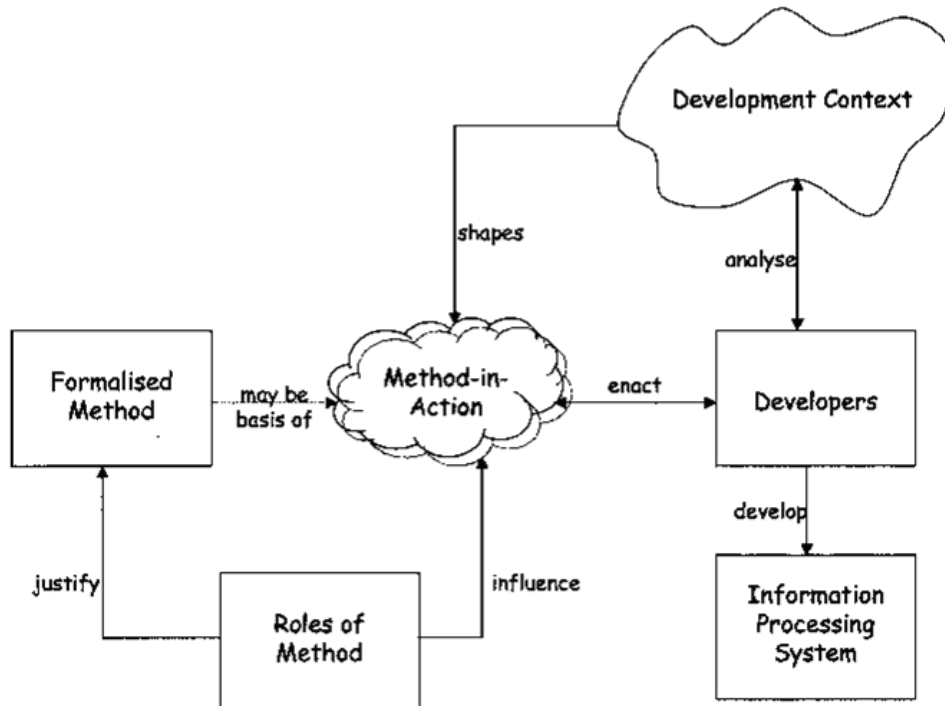
Begreper knyttet til systemutviklingsmetodikk blir benyttet om hverandre og visse metodologier gir de forskjellige rollene i prosjektet andre navn enn hva som er vanlig i forhold til andre praksiser. I for eksempel et Scrumprosjekt kalles en iterasjon i prosjektet for en sprint og en prosjektleder har navnet Scrum-master. Dette problemet er det forsøkt å få en oversikt over i dette underkapittelet. Tabellen begrepsdefinisjoner beskriver de begreper som gjelder for denne oppgaven. Når begrepene blir brukt i denne oppgaven er det forklaringen i denne tabellen som er gjeldende, med mindre noe annet er anvist ved bruk av begrepet. Selv om flere av begrepene presentert i oppgaven i praksis blir uttalt og benyttet i sin opprinnelige form på engelsk har jeg valgt å oversette de fleste uttrykk til norsk for å bedre lesbarheten av denne oppgaven. De oversatte begrepene står oppført med forklaring i begrepsdefinisjonene som vist i tabell 1.

Tabell 1 - Begrepsdefinisjoner

| Begrep | Forklaring |
|----------------|---|
| Agile | Engelsk for smidig |
| Akseptansetest | Test av krav kunden har satt for å se til at funksjonaliteten er i tråd med ønsket fra kunde. |
| Burn-down graf | Graf som viser hvor mye av arbeidsmengden som faktisk er utført, i forhold til hva som er planlagt. |
| CMS | Content Management System |
| Iterasjon | Avgrenset periode innenfor et prosjekt. |
| Produktbaklogg | Liste med funksjonalitet som ennå ikke er implementert i systemet. |
| Produkteier | Representant fra kunden, prioriterer produktbakloggen. |
| Scrum | Smidig systemutviklingsmetodikk, begrep hentet fra en scrum i et Rugbylag. |
| Scrum-master | Fasilitator for prosjektmøter og kommunikasjon mellom scrum-teamet, kunderepresentanter og ledelse. |
| Scrum-team | Prosjektgruppe sammensatt av personer med kompetanse til å løse et felles mål. |
| Smidig | Begrep om familien av systemutviklingsmetodikker basert på "Agile manifesto". |
| Sprint | En iterasjon i et scrumprosjekt |
| Sprintbaklogg | Liste med oppgaver som skal implementeres i løpet av en sprint |

2.2 Bruk og tilpasning av metodikk

Utviklingsmetodikken som faktisk blir benyttet er et resultat av flere faktorer. Fitzgerald (2002) viste med rammeverket method-in-action at den faktisk benyttede metodikken påvirkes av den formaliserte metoden, utviklingskontekst og systemutviklerne.



Figur 1: Rammeverk for metodebruk (Fitzgerald et al., 2002)

Fitzgerald(2002) sitt method-in-action rammeverk baserer seg på faktorer som påvirker, grunngrir og former den faktisk benyttede metodikken i organisasjonen. Method-in-action rammeverket (Fig. 1) viser ikke hvordan hvert enkelt prinsipp som omfattes av metodikken blir planlagt og benyttet i organisasjonen, men har et overordnet perspektiv i forhold til den helhetlige metodikken. For å beskrive de enkelte praksiser i en organisasjon kan NIPO rammeverket fra Larsen, Paivarinta og Smolander (2007) være et godt rammeverk. Dette rammeverket består av to dimensjoner, hvor bedriftens metodiske prinsipper plasseres i forhold til tiltenkt omfang av definert praksis og faktisk omfang av benyttet praksis. Prinsipper plassert i rammeverket beskrives ved hjelp av kategoriene (None)Ingen, Individuell, Prosjekt og Organisasjon i forhold til begge dimensjoner i rammeverket. Ved å kartlegge og plassere bedriftens praksiser i dette rammeverket, kan det bidra til et grunnlag og forklaring for analyse av bedriftens faktiske bruk av metodiske prinsipper.

Tabell 2 - Generisk NIPO rammeverk

| Faktisk omfang av benyttet praksis | Tiltenkt omfang av definert praksis | | | |
|------------------------------------|-------------------------------------|-------------|----------|--------------|
| | (None) Ingen | Individuell | Prosjekt | Organisasjon |
| (None) Ingen | | | | |
| Individuell | | | | |
| Prosjekt | | | | |
| Organisasjon | | | | |

I en ideell virkelighet hvor alle metodiske prinsipper ble benyttet som planlagt ville de metodiske prinsippene i NIPO rammeverket vært plassert på plasseringen O-O, P-P, I-I. Plasseringen N-N, beskriver en ikke benyttet praksis som ikke er definert, og er derfor ikke praktisk mulig å finne i noen prosjekter.

Det har også vist seg at metodikker som er laget for et foremål kan endre seg etter hvert som forståelsen av metodikken hos metodebrukerne skiller seg fra metodikkens opprinnelige ide (Päiväranta, Sein, & Peltola, 2007). Det er gjort flere forsøk på å veilede utviklingsteam til å gjøre de rette valgene av metodiske prinsipper basert på visse kriterier. Det er bant annet utviklet et ”verktøykasseprogram” som en prosjektleder kan kjøre for å tilpasse metodikken etter beste praksiser i forhold til et sett med kriterier. (Mnkandla & Dwolatzky, 2007) Alistair Cockburn, bakmannen bak den smidige familien av metodikker Crystal (Cockburn, 2002), har også definert et rammeverk som kan hjelpe til å definere et prosjekts metodikk ved å følge fire prinsipper til metodevalget; (1) Jo større prosjektgruppe, desto mer omfattende metodikk, (2) Jo mer kritisk systemet er, desto viktigere er det å benytte seg av en synlig og riktig konstruksjonsprosess. (3) En relativt liten økning i metodestørrelsen, legger til en relativt stor kostnad til prosjektet og (4) Den mest effektive formen for kommunikasjon er interaktiv og ansikt til ansikt, som ved en tavle (Cockburn, 2000).

Det er vist at prosjekter med hell kan benytte seg av praksiser fra flere forskjellige smidige metodikker. I en studie gjort av Fitzgerald, Hartnett og Conboy (2006) viste det seg at Intel Shannon hadde implementert praksiser kombinert fra både Scrum og Extreme Pogramming. Prosjektet hadde benyttet seg av Scrum for de ”myke” organisatoriske prinsippene, og hentet praksiser fra XP mot de mer ”harde” tekniske utfordringene.

Bedriftskultur og tilgjengelige ressurser har vist seg å ha en betydning på hvor enkelt det er å implementere enkelte systemutviklingspraksiser. I mange prosjekter kan det være en utfordring å implementere praksiser helt slik teorien sier, da organisasjonsstruktur og kultur kan være forskjellig fra det ideelle metodiske miljøet. En case studie for et webutviklingsprosjekt hvor prosjektet skulle implementere XP, viste det seg at det krevde en relativt stor tilpasning av organisasjonskultur for å implementere flere praksiser definert av metodikken (Grossman, Bergin, Leip, Merritt, & Gotel, 2004).

I en studie gjort hos 3 danske og 12 amerikanske systemutviklingsfirmaer ble det funnet faktorer som påvirker deres metodikk og kultur. Her førte tidspress, vage spesifikasjoner og mangel på erfaring til at organisasjonene tok i bruk praksiser som prototyping, parallell utvikling, fastsatt arkitektur og å fokusere på en hurtig løsning heller enn å fokusere på kvalitet (Baskerville & Pries-Heje, 2004).

2.3 Smidig systemutvikling

17 forkjempere for lettvekts systemutviklingsmetodikker samlet seg i Utah 2001. Sammen ville de bli enige om et sett med prinsipper de sammen var enige om. Dette er hva de kom frem til; resultatet ble kalt "The agile manifesto"

"Vi avdekker bedre måter å utvikle programvare på ved å hjelpe andre til å gjøre det. Gjennom dette arbeidet har vi kommet til å verdsette:

Individer og interaksjoner over prosesser og verktøy
Fungerende programvare over omfattende dokumentasjon
Kundesamarbeid over kontraktsforhandlinger
Reagere på endringer over å følge en plan

Det betyr, selv om det er verdi i begrepene til høyre, verdsetter vi begrepene til venstre mer." (Oversatt fra (Cockburn, 2002, s213))

De som skrev under dette manifestet er bakmenn for flere av de mest kjente smidige systemutviklingsmetodikkene blant disse: Crystal familien, Scrum, Extreme Programming, DSDM og Feature-driven Development med flere.

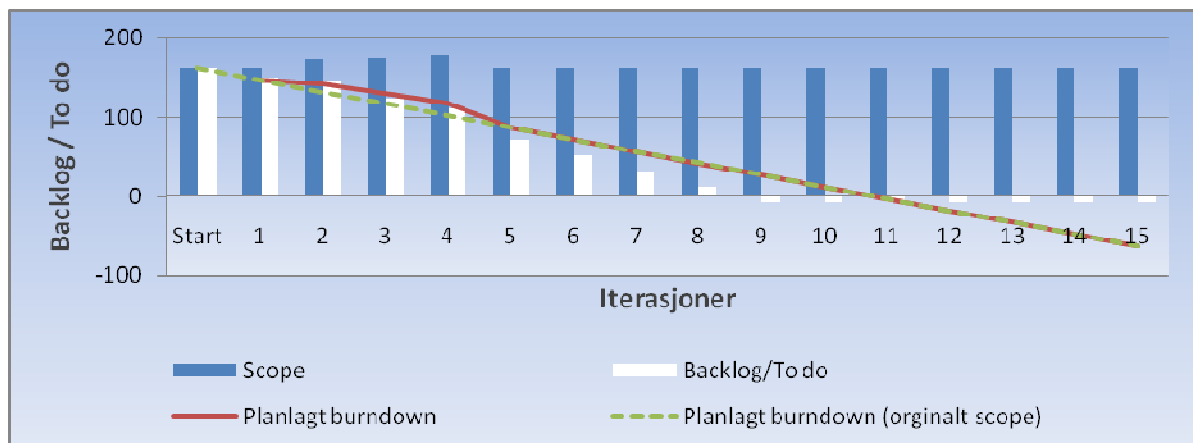
2.3.1 Smidige metodiske prinsipper

Prinsippet med å bruke brukerhistorier istedenfor omfattende designdokumentasjon kommer fra utviklingsmetodikken Extreme Programming hvor funksjonalitet tiltenkt systemet blir utformet i form av enkle kort som beskriver hvem som skal benytte funksjonaliteten, hva funksjonaliteten utfører og en begrunnelse for hvorfor denne funksjonaliteten skal implementeres. Bakgrunnen for denne måten å definere funksjonalitet på, er for å gjøre det enklere å kommunisere med brukeren av systemet om hva systemet skal gjøre uten å trenge å benytte seg av tekniske termer. (Waters, 2008; Wells, 1999)

Planning Poker er en teknikk utviklet foreløpig å bidra til at hele utviklingsteamet skal kunne delta i å estimere omfanget av et prosjekts brukerhistorier. Målet med teknikken er at utviklingsteamet kan komme til enighet etter å ha hørt hverandres argumenter, likt Delphi teknikken. En fasilitator leser opp en brukerhistorie og hver enkelt utvikler svarer med å vise et kort påskrevet en verdi. Dersom denne verdien varierer mellom utviklerne, skal det diskuteres hva som legges i den høye verdien av kompleksitet og det stemmes så på nytt. Det finnes flere forskjellige verdityper benyttet i Planning Poker. Haugen (2006) benytter seg av verdier på 0,5, 1, 2, 3, 4, og >4 som en verdi på hvor mange dager med par-programmering det vil benyttes til å utvikle den eventuelle oppgaven. Prosjektene denne oppgaven undersøker benytter seg av en teknikk som baserer seg på poeng. Skillet

mellom poeng og timer betyr at en ikke vet nøyaktig hvor mange timer en brukerhistorie tar å utvikle før en har utviklet noen. Underveis i prosjektet vil en oppnå en hastighet i prosjektet som tilsvarer et visst antall poeng pr iterasjon. Dette hastighetsmålet kan da gi et relativt nøyaktig estimat på lanseringsdag.

Prosjektarten kan illustreres som en graf kalt burn-down graf. Denne grafen viser prosjektets omfang i forhold til hvor mye arbeid er igjen å utføre og hvor mye arbeid som ble gjort de forrige iterasjonene. En estimert linje viser også hvor mange iterasjoner det er igjen før nåværende mengde gjenstående arbeid kan forventes å være ferdig utviklet.



Figur 2: Burn-down eksempel fra prosjekt 1

Figur 2 viser en slik burn-down graf hentet fra prosjekt 1 i denne undersøkelsen. De blå søylene presenterer det totale omfanget av systemet, mens de hvite søylene presenterer hva som er igjen å produsere. Den stiplede linjen viser den planlagte progresjonen i prosjektet i forhold til det utgangspunktet som var ved prosjektstart, mens den solide røde linjen viser den planlagte progresjonen i forhold til det faktiske omfanget av systemet etter eventuelle justeringer i omfang. Disse linjene går forbi nullpunkt, i en situasjon der det omfanget av systemet drastisk ville øke, ville den stiplede og solide linjen splittes og treffe null ved forskjellige iterasjoner. Noe som ville presentert hvor mange iterasjoner ekstra et eventuelt større omfang av systemet ville hatt i forhold til hva som ble avtalt ved prosjektstart.

Akseptansetester benyttes for å teste om utviklet funksjonalitet dekker det behovet kunden har sagt funksjonaliteten skal dekke. Typiske enhetstester dekker gjerne tekniske feil i programmeringskoden, mens akseptansetesten finner eventuelle logiske feil i systemet. Testen tar utgangspunkt i de oppgaver en eventuell brukerhistorie inneholder og skrives av en kunderepresentant før utviklerne har implementert programvaren (Miller & Collins, 2001).

At utviklingsteamet er samlokalisert betyr at kommunikasjonen kan foregå muntlig, ansikt til ansikt. Denne typen kommunikasjon regnes som den mest effektive måten å overføre kunnskap på og anbefales derfor i smidige utviklingsprosjekter. Terskelen til å forhøre seg med en kollega er vesentlig mindre dersom vedkommende sitter ved siden av,

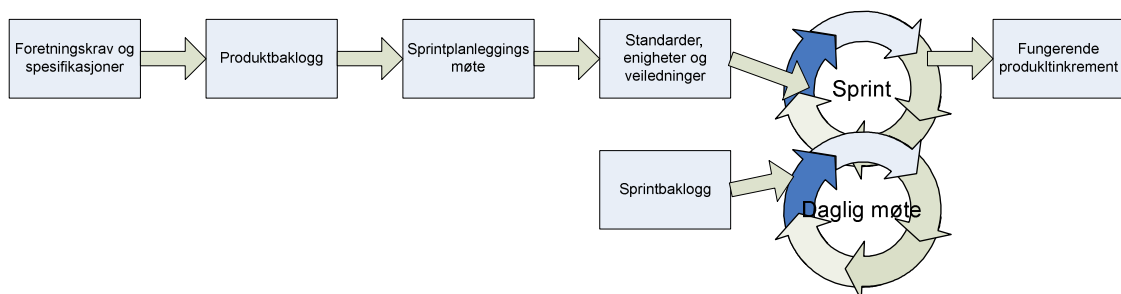
enn dersom utviklerne hadde måtte benytte seg av e-post eller telefon. Dette kan utnyttes ved å utforme prosjektlokalene og plassering av utviklere i henhold til å minske avstanden mellom utviklerne (Cockburn, 2002).

I et smidig prosjekt tar kunden en aktiv rolle i prosjektet. I forhold til Scrum har kunden ansvar for produktbakloggen og styrer prosjektets neste oppgaver ved hjelp av denne. I andre smidige metodikker som for eksempel XP, er kunden til stede for utviklerteamet til enhver tid. Dette medfører at kunden må få en forståelse for hvordan utviklerteamet arbeider og hvordan prosessen som benyttes fungerer. Martin, Noble og Biddle (2003) forklarer at det er visse egenskaper en kunderepresentant i et XP prosjekt behøver. Blant disse er forståelse for domenet, forståelse for at forretningsverdien programvaren som utvikles kan gi, kunne ta avgjørelser underveis og villighet til å ta ansvar for suksess eller nederlag. I organisasjonen undersøkt her, er det vanlig å ta en enkel gjennomgang av prosessen for kunden på et tidlig stadium, nettopp for å lære opp kunden.

2.4 Scrum

Systemutviklingsmetodikken Scrum baserer seg på et sett produktutviklingsprinsipper presentert av Hirotaka Takeuchi og Ikujiro Nonaka(1986). Gjennom analyse av den tids vellykkede innovasjonsbedrifter (blant annet Fujii-Xerox, Canon, Honda og 3M) fant de et sett med kriterier som viste seg å være felles for utviklingen av nye innovative produkter: Innebygget ustabilitet, selvorganiserende lag, overlappende utviklingsprosesser, ”multilæring”, subtil kontroll, og organisasjonell overføring av kunnskap. De beskrev den nye måten å arbeide på som et samarbeidende rugbylag som jobbet mot et felles mål.

I 1996 hjalp Ken Schwaber Individual, Inc. til å bli en mer fokusert og effektiv utviklingsbedrift ved hjelp av lærdom fra Takeuchi og Nonaka. De gode erfaringene fra denne første implementasjonen av Scrum, førte til en videre definering av praksissene benyttet i prosjektet (Schwaber & Beedle, 2002). Flere av disse praksissene kan knyttes direkte til ”The new product development game”, mens andre er tilpasset Schwabers erfaringer og kunnskap innenfor systemutvikling. Praksisene definert av Scrum innebefatter produkt baklogg, scrum-team, sprint, daglige møter, sprintplanleggingsmøte og sprintgjennomgangsmøte.



Figur 3: Scrumprosessen (Schwaber & Beedle, 2002, s8 (gjenprodusert))

Som vist i Figur 3 er en Scrum en iterativ prosess, hvor hver iterasjon (kalt Sprint) styres av en sprintbaklogg, denne bakloggen er et utdrag fra produktbakloggen som er en liste over all funksjonalitet i systemet. Sprinten som foregår over 30 dager styres av et daglig møte. Avsluttet sprint fører så med i et fungerende utviklet tillegg til programvaren som utvikles.

2.4.1 Griser og kyllinger

Et scrumprosjekt består i hovedsak av tre roller (Tabell 3). Scrum-team medlem, Scrum-master og produkteier. Rollenes oppgaver er beskrevet i underkapitlene, hvor jeg går nærmere inn på hver enkel praksis i Scrum.

Tabell 3 - Roller i et scrumprosjekt

| Rolle | Oppgave |
|-------------------|---|
| Scrum-team medlem | Bidra til å nå det mål som er satt for en sprint. |
| Scrum-master | Bidra til å fjerne de hindringer scrum-teamet måtte ha for å nå målet satt for en sprint. Fasilitere scrumprosess og møter. |
| Produkteier | Ta i mot ønsker til funksjonalitet Prioritere produktbaklogg Definere produktbakloggens oppgaver etter hvert som forståelsen for problemet blir bedre |

“A chicken and a pig are together when the chicken says, “Let’s start a restaurant!”

The pig thinks it over and says, “What would we call this restaurant?”

The chicken says, “Ham n’ Eggs!”

The pig says, “No, thanks. I’d be committed, but you’d only be involved!””

(Schwaber & Beedle, 2002, s 42)

Scrum skiller rollene i et prosjekt ved å kalle dem griser og kyllinger. Som historien over beskriver har grisene i et prosjekt dedikert seg helt og holdent til prosjektet, mens kyllingene er involverte av forskjellige årsaker. Griser i et scrumprosjekt beskrives som teammedlemmene inkludert scrum-master. Alle andre beskrives som kyllinger. Schwaber & Beedle (2002) beskriver at kyllingers involvering i prosjekter kan bidra til å skape støy, denne støyen kan for eksempel være innblanding i daglige møter, direkte kontakt med scrum-teamet for å fremlegge forslag til endringer i funksjonalitet. Støy hindrer scrum-teamet å dedikere seg selv til det målet som er satt for en sprint, og bør derfor reduseres.

2.4.2 Scrum-team

Scrum-teamet består av personene som skal nå et mål som er satt for en periode. Dette målet er bestemt i starten av hver iterasjon i sprintplanleggingsmøtet. Teamet er fasilitert av en person som tar rollen som Scrum-master, denne rollen fylles gjerne av en person

som ellers hadde tatt rollen som prosjektleder for prosjektet. Scrum-master sin hovedoppgave i prosjektet blir å sørge for at laget følger den prosessen som er satt, fasilitere prosjektmøtene; daglig møte, sprintplanleggingsmøte og sprintgjennomgang, og ”rydde vei” for teammedlemmene slik at det kan konsentrere seg om å nå målet istedenfor å bry seg om andre oppgaver. Teammedlemmene består av forskjellige individer gjerne med en variasjon i kompetanse som til sammen tilsvarer det som trengs for å nå sprintmålet som er satt. Laget står fritt til å løse oppgaven som er gitt og kan bruke den fremgangsmåten laget finner best for å nå målet til fastsatt tid. Da innenfor de retningslinjer, normer og regler som finnes for organisasjonen som en helhet. Schwaber (2002) beskriver den ideelle teamstørrelsen til å være syv personer, pluss eller minus to. For små lag vil påvirke produktivitetsgevinsten ved bruk av Scrum, og for store team vil kreve for mye tid i de daglige møtene, og kontrollmekanismene vanskeliggjøres. Større lag, anbefales å brytes ned flere mindre Scrum-team, som jobber mot den samme produktbakloggen. Det anbefales at medlemmene sitter samlet for å danne en god kommunikasjon innad i teamet og for å dele kunnskap.

2.4.3 Produktbaklogg

Produktbakloggen omfatter det arbeidet som gjenstår å utvikle i et prosjekt. All funksjonalitet, krav, problemer og behov skal beskrives og settes i en liste. Denne listen er prioritert og oppdatert av produkteieren kontinuerlig og er derfor et levende dokument gjennom hele systemets levetid. Selv om denne listen beskriver hele systemet, betyr ikke dette at den inneholder all den informasjon som kreves for å starte arbeidet med alle punktene i listen, men scrumteorien påpeker at denne bakloggens formatering skal stille til de forventningene som kommer fra organisasjonen. Det kan også finnes rene problemer som ikke er mulig å arbeide direkte med produktbakloggen, men disse defineres da som ”Ikke arbeidbare elementer” i lista. Det er helt og holdent opp til produkteieren å definere en løsning på et slikt problem, slik at scrum-teamet kan løse oppgaven. Siden denne produktbakloggen hele tiden er oppdatert og prioritert, kan scrum-teamet til enhver tid ha oversikt over hva kunden ønsker utviklet først, og har derfor en enkel jobb når det skal plukkes oppgaver som skal utvikles til et ferdig produkt.

2.4.4 Sprintplanleggingsmøte

Sprintplanleggingsmøtet har som mål å definere det målet scrum-teamet skal sette seg for den neste perioden (neste sprint). Møtet deles i to faser. Den første fasen deltar kunde, brukere, ledelse, produkteier og team og målet er å definere hva som skal produseres av funksjonalitet i neste sprint. Denne funksjonaliteten velges ut i fra produktbakloggen, hvor de oppgavene som er definert med høyest prioritet velges ut først. Den neste fasen består i at Scrum-teamet definerer hvordan de skal løse oppgavene satt i fase 1 og bygge en sprintbaklogg. Sprintbakloggen består av større og mindre konkrete oppgaver, som til sammen vil løse målet satt for sprinten.

2.4.5 Sprint

En sprint i Scrum, er en avgrenset periode i prosjektet hvor scrum-teamet jobber med et sett oppgaver de har valgt i sprintplanleggingsmøtet. Laget jobber kontinuerlig mot å nå målet som er satt og fraviker ikke fra dette målet. Sprintperioden er satt til 30 dager og det er ikke anbefalt å fravike fra dette. Allikevel viser det seg at flere gjør tilpasninger til

nettopp denne lengden. I en studie gjort på implementering av Scrum i 3M, viste det seg at produkteierfølte at 30 dager fravek fra å være smidig. Resultatet ble til at teamet besluttet å endre sin sprintlengde til to uker. (Moore, Reff, Graham, & Hackerson, 2007) Det eneste Scrum-teamet er pliktige til å forholde seg til i løpet av en sprint er denne sprintbakloggen og de daglige møtene. Scrum-teamet organiserer seg selv, og fordeler oppgaver og ansvarsområder etter som behovet melder seg.

Dersom det viser seg at vesentlige omstedigheter har endret seg i løpet av en sprint, kan en avslutte en sprint tidligere enn planlagt. Disse omstedighetene kan være seg endrede krav, omstedigheter, eller at teamet selv ser at det er for mange uvissheter rundt et mål satt for sprinten.

2.4.6 Daglige møter

Hver dag samles Scrum-teamet for et kort, men konkret møte. Schwaber (2002) gir møtet et tidsperspektiv på 15 minutter, og i løpet av disse minuttene skal Scrum-master spørre hvert enkelt scrumteam-medlem hva han eller hun har gjort siden siste daglig møte, hva han eller hun skal gjøre frem til neste daglig møte og hva som hindrer han eller hun i å gjøre arbeid. Teamet skal, i den grad det er mulig, være samlet til disse møtene. Dersom det er eventuelle problemer med å kunne møte, kan eventuelle perifere teammedlemmer delta på møtet over telefonkonferanse. Det er også åpent for at andre interessenter kan delta på disse møtene, da deltagelse på disse møtene gir verdifull innsikt i status på prosjektet. Det påpekes derimot at tilhørere som ikke er en del av scrum-teamet ikke skal delta aktivt i samtalene, men ta del i møtet som en passiv tilhører. Eventuelle diskusjoner som kommer opp i løpet av et daglig møte, skal ikke diskuteres direkte under møtet, de partene som måtte ønske å diskutere en sak videre, vil da måtte samles i etterkant av møtet for å kunne diskutere videre da.

2.4.7 Sprintgjennomgangsmøte

Sprintplanleggingsmøtet er et møte hvor scrum-teamet gjennomgår sammen med ledelsen, produkteieren og kunden hva de faktisk har produsert i løpet av en sprint. Møtet skal begrenses til fire timer og presentasjonen skal beskrive sprinten på enklest mulig måte. Schwaber (2002) nevner at selv PowerPoint presentasjoner burde forbyes som et verktøy for denne presentasjonen, da det både krever større forberedelse som tar konsentrasjon til teammedlemmene bort fra produksjon. Målet med møtet er å øke alle interessers forståelse for systemet, og kunne peke en vei videre til neste sprint.

3 Metodebeskrivelse

3.1 Forskningsdesign

For å løse problemstillingen har jeg i denne oppgaven er en kvalitativ tilnærming. Dette fordi jeg ønsker å gå i dybden på et enkelt firma og slik forklare den virkeligheten som eksisterer der. I den overordnede forskningsmodellen har jeg hentet inspirasjon fra Jacobsen(2005).



Figur 4: Forskningsmodell

3.2 Datainnsamling

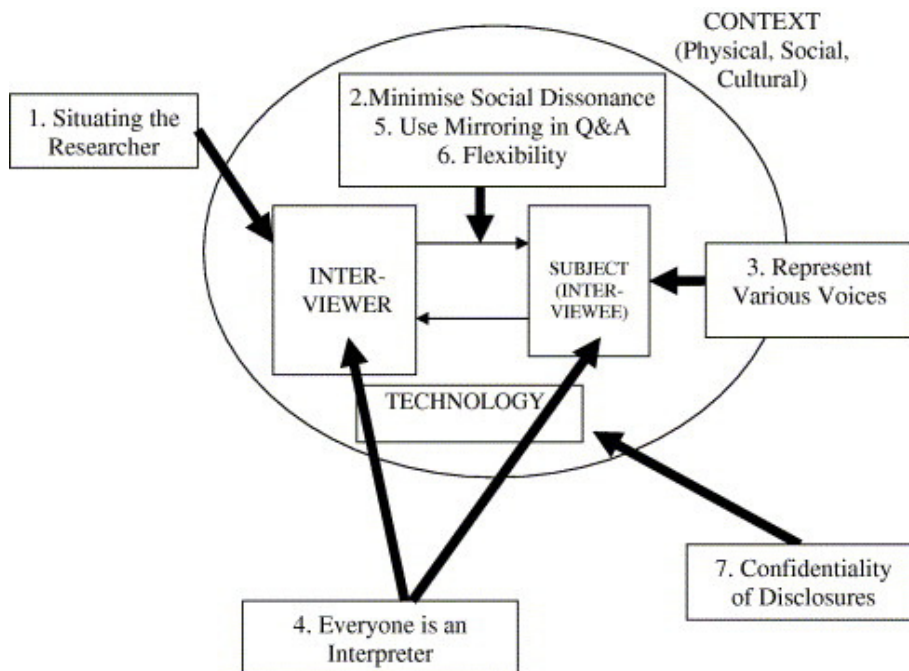
Data innsamlet til denne oppgaven stammer i hovedsak fra intervjuer med prosjektdeltagere i de utvalgte prosjektene. Blant disse er prosjektledere eller Scrum-master den sterkeste kilden til data som omhandler valg i forhold til utviklingsprinsippene i prosjektene, i et av prosjektene er også prosjektdeltagere i form av systemutviklere intervjuet for å kunne bidra til en mer nyansert fremstilling av prosjektene. Sammen med data hentet fra intervjuene er det foretatt observasjoner av det daglige arbeidet, samt en gjennomgang av prosjektdokumentasjon som er skrevet i prosjektene.

Tabell 4 - Oversikt over datainnsamling i prosjektene

| Prosjekt | Intervju | Observasjon | Dokumentasjonsgjennomgang |
|------------|----------------------------------|-------------|---------------------------|
| Prosjekt 1 | Scrum-master, 2x systemutviklere | Ja | Ja |
| Prosjekt 2 | Scrum-master | | |
| Prosjekt 3 | Scrum-master | | Ja |
| Prosjekt 4 | Scrum-master | Ja | |

3.2.1 Intervju

Intervju som datainnsamlingsmetode er en veldig gunstig måte å samle inn data på når det er relativt få enheter som skal undersøkes, når vi er interesserte i hva hvert enkelt individ sier og når vi er interessert i hvordan den enkelte tolker fenomenet(Jacobsen, 2005). I dette tilfellet har det vært ønskelig å undersøke hvert enkelt prosjekt, og undersøke hvilke valg, og hvorfor disse valgene er tatt i forhold til valgene av metodiske prinsipper. Intervjuene er foretatt som semistrukturerte intervjuer, hvor temaer, istedenfor spørsmål styrer intervjuet fremover (Intervjuguide vedlegg 1). Det har blitt lagt vekt på Myers & Newman (2007) veiledning til den dramaturgiske modell for det kvalitative IS intervjuet (fig 5) ved både forarbeid og under intervjuprosessen.



Figur 5: Veiledning til det kvalitative forskningsintervju (Myers & Newman, 2007)

Denne veiledningen beskriver syv punkter som er anbefalt å følge under et kvalitativt intervju. (1) For å sette situasjonen kan det være en fordel å forklare hvem jeg som intervjuer er, hva slags bakgrunn jeg har og bakgrunnen for undersøkelsen. Dette er tatt høyde for i intervjuene både ved at jeg som intervjuerperson uformelt prater om oppgaven og min setting, og ved at respondentene allerede kjenner meg fra deltagelse som observatør i prosjektene. (2) Minimere sosial dissonans vil si å fjerne eventuelle faktorer som kan gjøre respondenten ukomfortabel, noe som ble ivarettatt ved å sikre konfidensialitet, gjennomføre intervjuet på respondentens "hjemmebane" og å forklare hvordan dataene skulle bli benyttet i oppgaven. Intervjuene har også i flere tilfeller blitt gjennomført etter en liten uformell prat slik at begge parter er trygge på omgivelsene. (4) Alle parter tolker situasjonen, i noen tilfeller kan respondenten ha følt at jeg som intervjuer prøver å finne feil. Dersom en slik situasjon har oppstått, er det viktig å betrygge om mitt mål med oppgaven og at jeg ikke er ute etter å "ta" noen. (5) Bruk av speiling av spørsmål kan bidra til å få dypere forklaringer på interessante temaer og har blitt aktivt benyttet med hell i flere av intervjuene. (6) Fleksibilitet er ivarettatt både for å ha muligheten til å grave dypere i interessante temaer, og eventuelt dersom det er aktuelt å endre samtaleemne dersom noe enten er uinteressant, eller klart ubehagelig for respondent å prate om. Intervjuguidens struktur har også ivarettatt fleksibiliteten i intervjuene. (7) Konfidensialitet betyr at data og transkripsjoner hentet fra intervjuene ikke skal være tilgjengelig for andre enn de som benytter seg av dataene. Alle intervjuer, navn og bedriftsnavn er anonymisert i løpet av prosessen, for å ivareta dette punktet.

Samtlige intervjuer er gjennomført ansikt til ansikt og med bare respondent og intervjuer til stede i rommet. Samtalene er tatt opp på lydbånd ved hjelp av mobiltelefonen Nokia E51 sin opptakerfunksjon. De digitale lydfilene er lastet inn i programmet Transcriber,

som gjør det mulig å stykke opp intervjuet i seksjoner for transkribering (Manta, Antoine, Galliano, & Barras, 1998). Rådata fra intervjuene er vedlagt under kapittelet vedlegg.

3.2.2 Observasjon

Ved å være fysisk tilstede i produksjonslokalene vil jeg som observatør kunne vurdere adferden til prosjektteamet. Jacobsen (2005) presiserer at det er en risiko for at personer som blir observert vil endre sin adferd når de blir observert. Dette ble tatt høyde for ved å fysisk plassere meg på en slik måte at jeg ikke ville være i direkte kontakt med prosjektdeltagerne, samt at jeg deltok med en såpass hyppighet at prosjektteamet fant det naturlig at jeg var til stede. Funn under observasjon ble notert, og senere transkribert. Av tidsmessige og geografiske hensyn har det kun blitt foretatt observasjonsstudier av prosjekt 1 og en sesjon i prosjekt 4. Enkelte observasjoner er gjort på vei mellom prosjekter, da flere av prosjektlederne var til stede i organisasjonens lokaler. Disse observasjonene har styrket min forståelse for fenomenet, men i mindre grad blitt notert ned som data til denne oppgaven og må derfor bare ha en begrenset verdi i forhold til datainnsamlingen.

Gevinsten med å observere prosjektlagene er at jeg som forskeren kan bekrefte eller avkrefte bruk av metodiske prinsipper i praksis. I flere intervju har jeg erfart at intervjuobjektet ønsker å prate om hvordan visse typer problemer skal løses i teorien, istedenfor å diskutere hva som faktisk blir gjort i prosjektene. Observasjon av prosjektteamene har da bidratt til at jeg som forsker forstår bedriften prosjektenes situasjon i større grad enn dersom oppgaven kun var basert på intervjudata.

3.2.3 Gjennomgang av prosjektdokumentasjon

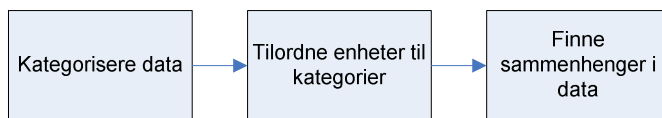
Gjennomgang av dokumenter kan være nyttig når en vil vite hva personer faktisk har gjort, vedtatt og sagt tidligere. Dessverre er det slik at denne informasjonen er hentet inn og tilpasset et annet foremål enn hva undersøkelsen omfatter (Jacobsen, 2005). I dette tilfellet har det vært aktuelt å følge prosjektplanen se på sammenhengen mellom avtalte praksiser og gjennomføringen. Bedriften benyttet også et exel-skjema for å holde rede på prosjektenes baklogg og hvordan prosjektet beveger seg fremover. Dette skjemaet har gitt god informasjon om hvordan bedriften lager sine brukerhistorier, fremgang og endringer i prosjektene underveis. Det har også blitt gitt tilgang til flere av statusrapportene i prosjekt 1, hvor kunde blir oppdatert på prosjektstatus. Selv om disse dokumentene ikke har gitt veldig mye ny informasjon om prosjektene ut i fra hva som er observert og fått frem gjennom intervjuer, bidrar denne dokumentasjonen til å gi en forklaringsgrad og en bedre forståelse om emnet.

3.2.4 Litteraturgjennomgang

Opgaven har tatt utgangspunkt i eksisterende forskning og litteratur innenfor problemområdet, og det er gjort i flere av de større nettbaserte artikkeldatabaser og søkemotorer. Google Scholar og Bibsys MIME er de søkemotorene som er benyttet i størst grad til litteratursøket da disse søker gjennom flere forskjellige journaler. Det er også blitt benyttet lærebøker fra masterstudiet Informasjonssystemer på Universitetet i Agder, da henholdsvis (Cockburn, 2002) og (Fitzgerald et al., 2002).

3.3 Dataanalyse

Utgangspunktet for dataanalysen var Grounded Gheory beskrevet av Strauss & Corbin (1998). Denne metoden baserer seg på å danne teorier på bakgrunn av et datagrunnlag og følger tre steg for å gjøre dette. Åpen koding, aksial koding og selektiv koding. Sammen med Grounded Theory har også Jacobsen (2005) blitt benyttet som guide til analysefasen av undersøkelsen, og stegene presentert der er som beskrevet i figur 6.



Figur 6: Dataanalyse modell

3.3.1 Kategorisere data

På bakgrunn i problemstillingen hvor undersøkelsen skulle undersøke hvilke prinsipper som ble benyttet i den undersøkte Scrumprosjekter, tok jeg først tatt utgangspunkt i de utviklingsprinsipper som finnes i Scrum litteraturen fra Schwaber og Beedle(2002). De praksisene som er presentert der, er grunnsteinene til bruk av Scrum og er utgangspunktet for alle prosjektene. Etter at hovedkategorier basert på litteraturgjennomgangen ble identifisert ble intervjutranskripsjonene gjennomgått. Metoden for den videre kategoriseringen foregikk ved å gjennomlese utskrift av transkripsjonene og markere utsnitt med tekst som handlet om samme tema med fargekoder. Denne analysen av transkripsjonene førte også til at flere kategorier ble lagt til, etter hvert som flere metodeprinsipper dukket opp. Min problemstilling leter etter svar på bakgrunnen til bruk av disse prinsippene og bakgrunnen til en eventuell tilpasning av disse.

3.3.2 Tilordne enheter til kategorier

Etter kategorisering av data ble det foretatt en andre gjennomgang av intervjutranskripsjonene, avsnitt og sitater som kunne plasseres innenfor kategoriene ble plassert enten som hele sitater eller som stikkord under kategoriene. Dette arbeidet ble gjort noe mer uformelt enn hva forskningsmetodikken sier, allikevel gav dette en god oversikt over de forskjellige funnene for det videre arbeidet med datagrunnlaget.

3.3.3 Finne sammenhenger i data

Jacobsen (2005) beskriver to måter kategorier kan ha sammenheng på:

1. Sammenhengen kommer fra at forskeren eller datagrunnlaget forbinder kategoriene med hverandre
2. Kategoriene kan ha en påvirkende effekt på hverandre

I denne oppgaven var målet å finne hva som faktisk påvirket valget og tilpasningene til bruken av systemutviklingspraksiser og det var derfor om å gjøre å finne hvilke kategorier og begreper som hadde en påvirkning på prinsippene funnet fra litteraturen og de prinsippene som var unike for organisasjonen som ble undersøkt.

Min fremgangsmetode ble da å gjennomgå kategoriene og se hva som var til felles for de prosjektene som hadde lik type bruk av praksiser, og da også prøve å finne de tingene som var unike med de prosjektene som hadde avvik i forhold til bruk av sin metodikk.

4 Resultater

4.1 Beskrivelse av organisasjon

Den undersøkte er en liten til middels stor bedrift på 60 ansatte med konterer i Kristiansand, Drammen og Oslo. Av kjerneområder har bedriften spesialisert seg på drift av it-systemer og konsulenttjenester innenfor systemutvikling, integrasjon, forvaltning og rådgivning.

Selv om flere av de store verdensomspennende konsulentselskapene baserer sin systemutviklingsmetodikk i stor grad på strukturerte og rigide metodologier har bedriften ledelse de siste årene valgt å markedsføre seg med at deres systemutviklingsprosjekter er organisert ved smidig utviklingsmetodikk, da med metodikken Scrum som basis. Ved å selge seg inn med smidig metodikk, håper ledelsen å kunne oppnå et konkurransefortrinn ved å tilby bedre kontakt med kunden i prosjektet, og oppnå bedre effektivitet internt.

Siden det er besluttet at bedriften skal benytte seg av Scrum som basis for sine prosjekter, har de også sendt flere prosjektledere på Scrum-master kurs, samtidig som flere av prosjektlederne har deltatt på prosjekter som har benyttet seg av Scrum tidligere.

4.2 Beskrivelse av prosjekter

Data innsamlet om de forskjellige prosjektene beskrives i dette underkapitlet. Funnene i prosjektene er kategorisert basert på fra de metodiske prinsippene definert i litteraturen om Scrum som beskrevet av Schwaber og Beedle (2002).

4.2.1 Prosjekt 1 – Reiselivsportal

Dette prosjektet var et mindre prosjekt, som går over et tidsrom på 3-4 måneder. Målet med prosjektet var å utforme en internettportal for å reklamere for flere interessenter i turistnæringen. Prosjektdeltagerne var et team satt sammen av to designere fra tredjeparts firma og en utvikler samt prosjektleder fra organisasjonen undersøkt i denne oppgaven. Prosjektet bygger på en CMS bedriften har benyttet tidligere, slik at programmeringsarbeidet består i å designe nettsidene grafisk, knytte seg opp mot eventuelle samarbeidspartner og tilpasse moduler for innholdspresentasjon.

Scrum-team

Prosjektet er relativt lite og har følgende struktur:

Tabell 5 - Prosjekt 1: Prosjektdeltakere

| Rolle | Organisatorisk tilhørighet |
|----------------|----------------------------|
| Scrum-master | Bedriften |
| Systemeier | Kunde |
| Systemutvikler | Bedriften |
| Systemutvikler | Samarbeidsbedrift |
| Webdesigner | Samarbeidsbedrift |

Scrum-teamet har plassert seg i kundens lokaler, i direkte tilknytning til produkteier i prosjektet. Denne plasseringen har gjort det lettere for prosjektet å holde kontinuerlig kontakt med produkteieren slik at eventuelle uklarheter kan bli forklart raskt. På spørsmål om hva denne kommunikasjonen med kunden bestod i, svarte den ene systemutvikleren slik:

”De spør om endringer, ting som de ser som de ser som de vil ha litt annerledes. Eller ideer og den type ting.”(Systemutvikler 1, prosjekt 1)

Denne kontakten mellom produkteier og scrum-team oppfattes som verdifull av scrum-teamet og de føler de ikke trenger å bekrefte avgjørelser med Scrum-master ved enhver endring..

Produkt baklogg

Produktbakloggen administreres ved hjelp av prosjektstyringsverktøyet. I produktbakloggen ligger all tiltenkt funksjonalitet beskrevet som brukerhistorier og disse er prioritert basert på en skala fra 1-3, hvor 1 er høyeste prioritet. Prioriteringene i produktbakloggen gjøres av kunden, og det er helt opp til kunden å prioritere gjennom hele prosjektet. På spørsmål om det faktisk skjer omprioriteringer underveis svarte en systemutvikler i prosjektet slik:

”Ja, det skjer det, og det står kunden helt fritt til å gjøre. Hele veien. Så at det som var høyt prioritert en uke, kan bli mindre prioritert neste gang.”(Systemutvikler 1, prosjekt 1)

Brugerhistoriene som er listet i produktbakloggen følger en standard mal som til enhver tid er tilgjengelig på scrum-teamets tavle. Denne malen beskriver brukerhistoriesyntaksen slik: ”Som <Rolle> skal jeg <oppgave> slik at <verdi>”. Allikevel oppfylte ikke 11 av 57 brukerhistorier sine egne krav om at alle brukerhistoriene skulle forklare verdien den ga systemet. Flere av brugerhistoriene som ikke har verdibegrepet med, har dette spesifisert i et beskrivelsesfelt, noe som kan tyde på at det er klarhet i den faktiske verdien og at det rett og slett er den skriftlige formuleringen som er vanskelig å få til i forhold til malen. I andre tilfeller, hvor ingenting er beskrevet, er det uklart å se verdien av brugerhistorien i forhold til systemet.

I prosjekt 1 er det også Scrum-master som administrerer produktbakloggen, produkteier får ukentlig en statusrapport som beskriver fremdrift med status, endringer og de oppgaver fra produktbaklogg det ble besluttet å ta med i sprintplanleggingsmøtet.

Sprintplanleggingsmøte

Sprintplanleggingsmøtet i prosjektet besto i å velge ut hvilke oppgaver som skulle produseres denne sprinten. I de tilfeller der det var oppgaver som ikke ble ferdige denne sprinten, ble oppgavene automatisk tatt med i den neste. På spørsmål om hva som skjer dersom en oppgave ikke var ferdig forrige sprint, fikk jeg følgende svar:

”De brukerhistoriene som ikke blir gjennomført på den planlagte iterasjonen, de blir fulgt med til neste uke. Så ingenting blir flyttet ut av systemet før det godkjent, det er testet av oss, kunden har sett på det, testet det og godkjent det. Og da blir det satt til "verifisert" på tavla.” (Systemutvikler 1, prosjekt 1)

Scrum-master forklarte denne praksisen slik:

”Så hvis ikke alle oppgavene til en brukerhistorie er tatt, så går brukerhistorien tilbake i prosjekt backlog. Og så må den eventuelt på mandagen, da prioriteres opp igjen, for å ta den siste oppgaven. Det er ikke alltid vi har gjort det like rigid. Det vil si, hvis vi vet vi skal ta en brukerhistorie og gjøre den ferdig, så gidder ikke jeg å ta den ned fra tavla på fredag ettermiddag, og henge den opp igjen på mandags morgen.”(Scrum-master, prosjekt 1)

Hva som hadde skjedd dersom denne brukerhistorien hadde blitt nedprioritert i løpet av sprinten kommer ikke frem fra datagrunnlaget, så det er usikkert om det hadde hatt konsekvenser for denne praksisen.

Sprint

Sprintlengden i reiselivsportalprosjektet er satt til en uke. Dette forklares med at kunden skal kunne gi tilbakemeldinger på resultatene hyppigere. Scrum master på prosjektet forklarer det slik:

”En ukes iterasjonstid gir kunden større mulighet til å påvirke prosessen. For i Scrum så er det sånn at kunden kan påvirke i iterasjonsplanleggingsmøtene. # Og jo kortere tidsfrist du har, jo kortere iterasjoner pleier du å ha.” (Scrum-master, prosjekt 1)

Målet med en ukes sprintlengde blir derfor for å ta høyde for at prosjektet er kort, og samtidig opprettholde hyppige retningsvalg.

Daglige møter

Prosjekt 1 benytter seg av prinsippet med daglige møter. Disse daglige møtene gjennomføres selv uten Scrum-master til stede, og har også kunderepresentanten tilgjengelig som aktiv deltager av møtet. Ved å ta med kunden, gir prosjektet dette daglige møtet en ny funksjon i tillegg til å informere Scrum-master om eventuelle hindringer i programmeringen. Kunden blir dermed en aktiv part i utviklingen i større grad enn hva Scrum sier om dette. En systemutvikler uttrykker sine erfaringer om det daglige møtet slik:

”Det synes jeg er greit, det er såpass kort også, at det har ikke noe å si. Det er greit å komme i gang hver dag kl. 9 # Det blir litt søkt, når standupen bare er meg og de to andre, sånn som det var i dag, da har vi bare gått gjennom hva vi hadde gjort, men når kunden og prosjektleder er der så er det veldig lurt.” (Systemutvikler 2, prosjekt 1)

Erfaringen med å ta inn kunden som deltager i de daglige møtene er positive. De utfordringene som blir tatt opp på de daglige møtene er ofte direkte knyttet til programmeringsrelaterte spørsmål, det ble presisert at Scrum-master ved flere anledninger hadde satt dem i kontakt med resurspersoner i og utenfor firmaet for å løse et problem. Det har hendt at det ikke har vært nok tilgjengelige folk til å gjennomføre et daglig møte, i disse tilfeller ble møtet utelatt den dagen.

Møtet er tidsavgrenset til 15 minutter, og deltagerne har mulighet til å diskutere problemer som måtte dukke opp i løpet av disse 15minuttene, prosjektdeltagerne står oppreist hele møtet, noe som har en naturlig begrensende effekt på lengden av møtet. Det ble observert at det daglige møtet enkelte ganger kun ble benyttet til å oppdatere hverandre og seg selv på hva de faktisk hadde gjort siden forrige daglig møte.

Sprintgjennomgang

Sprinten avsluttes med et sprintgjennomgangsmøte hvor iterasjonens gang blir gjennomgått.

Det har vist seg at det er et problem å få en brukerhistorie godkjent fra kunden, da akseptansekriteriene for en brukerhistorie ikke er klare når utviklerne påbegynner arbeidet med iterasjonen. Når utviklerne og kunde samles på iterasjonsplanleggingen, velges brukerhistorier og det genereres oppgaver, men disse trenger nødvendigvis ikke å samsvare med hva kunden spesifikt ønsker å oppnå. Det antydes at det har kommet akseptansekriterier fra kunden så sent som dagen før planlagt akseptansetest, noe som har ført med seg at testen feilet. En systemutvikler mente dette kunne føre til at sprintende har glidd i hverandre.

”Også dette med at akseptansekriteriene har kommet seint, de har jo komme midt i iterasjonen, så sånn sett så har det glid litt over hverandre. Men vi holder jo flyten, så, det er bra.”(Systemutvikler 2, prosjekt 1)

Den samme systemutvikleren påpekte også at de sene akseptansekriteriene medførte nye oppgaver. Disse oppgavene skulle løses innenfor samme ramme med tid, selv om de ga merarbeid for utviklingsteamet den aktuelle iterasjonen. Slik hadde prosjektet opparbeidet seg en situasjon hvor de ikke klarte å levere planlagte brukerhistorier til slutten av iterasjonen, siden det ikke alltid var brukerhistorier å teste siste dag, ble det heller ikke gjennomført akseptansetester hver iterasjon. Det var ikke definert noen regler for utforming og levering av akseptansekriteriene, verken skriftlig eller muntlig i forkant av prosjektet.

”Det eneste som varierer litt er når vi har kjørt akseptansetester, når kunden er med. Det har variert litt på når kunden er har mulighet og sånn. Det var egentlig noe som... jeg vet ikke helt på at det skulle være fast eller ikke, men det har vært litt sånn når kunden har vært klar, og når vi har vært klar med koden. Når vi har tatt en del oppgaver som var klar til test og kunden har skrevet kriteriene så har vi tatt en bolk her og der.”(Systemutvikler 1, vedlegg X)

Akseptansetesten ble gjennomført når teamet hadde utført nok brukerhistorier til at de selv syntes det var på tide å sette seg ned for å teste dem med kunden. Kundens tid til å gå gjennom akseptansetesten hadde også en innvirkning på når akseptansetesten ble gjennomført. Sitatet over antyder også at det eksisterte en usikkerhet hos utviklingsteamet om hva som faktisk var planen for gjennomføring denne prosessen. Som nevnt tidligere, viser det seg at ikke alle brukerhistoriene følger malen. Dette kan ha ført til de må utvides i form av utvidete akseptansetestkriterier. Under observasjon av et iterasjonsplanleggingsmøte, ble alle brukerhistoriene gjennomgått i plenum med kunderepresentant til stede. På dette møtet ble det ikke observert noen endringer eller spesifiseringer av krav i forhold til de planlagte oppgavene fra kunderepresentanten.

Det har også oppstått et problem ved at rolleoppgaver som er avtalt, formelt eller uformelt, ikke blir gjennomført av en av partene. Produkteier i prosjektene undersøkt i denne oppgaven har et stort ansvar for at prosessen skal fungere slik den ble planlagt. Det uttrykkes en usikkerhet fra scrum-master om hvordan denne situasjonen skal takles og påpeker at det er kunden som faktisk sitter med siste ord. Scrum-master i et av prosjektene uttrykker det slik:

”... men vi må fremdeles også huske at det er kunden som betaler lønna vår. For vi tilpasser oss selvfølgelig kunden.”(Scrum-master, prosjekt 1)

Dersom det er slik at kunden ikke prioriterer eller har tid til å utføre sin rolle fullstendig en uke, vil dette ha konsekvenser for prosjektet. Da det er kunden som betaler for bedriftens tjenester, er det kunden som sitter med størst makt i forhold til dette. Resultatet har vært at scrum-master må forklare konsekvenser i forhold til hva som skjer dersom et valg om avvik fra rollen til produkteier forekommer. Blant annet at akseptanskriteriene kommer senere enn ønsket. Scrum-master hadde forklart til kunden at konsekvensen av for sene akseptanskriterier, medførte at oppgavene ikke ble påbegynt.

”Hvis det ikke er inne på tirsdag, så begynner vi ikke på de. Så kan kunden velge. Men på en annen side, så er det kunden som styrer dette. Så hvis kunden sier at en skal drite i det og godkjenner det, hva skal en gjøre da?” (Scrum-master, prosjekt 1)

Denne situasjonen ville nødvendigvis medføre problemer med å få gjennomført akseptansetesten den gjeldende sprinten.

Andre praksiser funnet i prosjektet

Et prosjektstyringsverktøy basert på en Excel mal definert for logistikksystemprosjektet har blitt tatt i bruk i dette prosjektet også. All funksjonalitet i dette dokumentet er derimot ikke benyttet da Scrum-master ikke mener det passer helt inn i forhold til dette prosjektet. Allikevel er dokumentet benyttet til produktbaklogg, estimering av prosjektfart, prosjektstatus og burn-down, tilgjengelige resurser, og sprintbaklogg. Scrum-master beskriver bruken av Excel prosjektstyringsverktøyet slik:

”Excel arket er greit på noen ting, men det er ikke så veldig greit på andre ting. Så selv om du tar det som et utgangspunkt, så vil alle prosjektlederne her finne sin måte å gjøre ting på.”(Scrum-master, prosjekt 1)

Estimering av omfang på brukerhistoriene gjøres i poeng, og blir nevnt at dette er noe alle prosjekter skal gjøre.

”Vi estimerer alltid i poeng og poengene timeestimerer vi også og ganger det med kroner. Alle gjør sånn, og alle skal gjøre sånn. Så det er ingen annen måte å gjøre det i denne bedriften på.”(Scrum-master, prosjekt 1)

Prosjektet baserer seg på en enkel kodestandardisering, og følger noen faste regler.

”Ja, vi har retningslinjer for hva som er ferdig kode, at det skal følge kodestandard, være kommentert og alt skal kompilere. Og det skal være enhetstestet.”(Systemutvikler 1, prosjekt 1)

Det ble derimot ikke funnet noen dokumentasjon å på denne kodestandarden i prosjektet.

4.2.2 Prosjekt 2 – Prosjektstyring hos teknologibedrift

Prosjekt 2 var et utviklingsprosjekt hos en kunde. Bedriftens representant deltok som scrum-master, og utviklerne var satt sammen fra kunden sin utvikleravdeling sin stab. Prosjektet skulle ta form som en pilot på innføring av smidig metodikk i firmaet, hvor målet var å utvikle en gammel teknologisk løsning på ny plattform. Praksisene benyttet i dette prosjektet har derfor blitt påvirket av en annen bedriftskultur enn den undersøkte organisasjonen. Selv om dette blir et internt utviklingsprosjekt hos kunden, er det praksisene og erfaringene fra den undersøkte organisasjonen og Scrum-master som ligger til grunne for metodebruken i dette prosjektet. Scrum-master for dette prosjektet beskriver scrum-teamet som veldig klare for å ta i bruk Scrum som prosess i sine prosjekter. De har tidligere benyttet seg av en type prosess som i større grad baserer seg på stegvis utvikling, med veldig fokus på detaljanalyse av problemområdene og virker nå positive til å effektivisere seg.

Scrum-team

I kundeorganisasjonen er det ingen som har full oversikt over det systemet som skal leveres. Derfor er det heller ingen fullverdig produkteier. Derfor blir det en utfordring å få tatt en avgjørelse med tanke på hvordan løsningen skal bli. Den som innehar den offisielle rollen som produkteier i prosjektet, tar seg ikke av de oppgavene som er definert i Scrum. Produktbakloggen holdes av en kunderepresentant som senere skal fungere som scrum-master, mens avgjørelsene blir tatt av produkteieren. Dette beskrives som en utfordring, da scrum-master hele tiden må stille kontrollspørsmål og opplyse om eventuelle konsekvenser av avgjørelser. Produkteieren benytter seg også av en kollega som ikke er deltager i prosjektet for å hjelpe til med prosessen. Scrum-master påpeker at kunden i dette tilfellet ikke er interessert i selve IT-prosessen, og at det kan være grunnen til at produkteier ikke kan ta alle beslutninger. Prosjektets deltagere blir da som i følge tabellen under.

Tabell 6 - Prosjekt 2: Prosjektdeltakere

| Rolle | Organisatorisk tilhørighet |
|----------------------|-----------------------------------|
| Scrum-master | Bedriften |
| Produkteier | Kunde |
| Produkteierassistent | Kunde |
| Prosjektdeltakere | Kunde |

Produkt baklogg

Det er bevisst gjort en endring med tanke på produktbakloggen i dette prosjektet. Scrum-master forklarer det slik at kundens ansatte har "sine hatter" i prosjektet. De som driver med en type arbeid, jobber kun med det.

Det var det ikke hensiktsmessig i å lære opp produkteieren til å ta ansvaret for produktbakloggen i dette prosjektet, og Scrum-master forklarer det slik:

"Vi har en produkteier og etter i prosessen, skal han ha ansvar for den backloggen. Det har han ikke forutsetninger til å kunne gjøre i det hele tatt. Og med tanke på at han er bare produkteier for dette lille prosjektet, så er han ute, så kan det godt hende det går to år at han aldri vil være i den rollen igjen. Så er det ikke hensiktsmessig å trene han opp for å kunne ta fullt eierskap på den eller det vil si administrasjon." (Scrum-master, prosjekt 2)

Resultatet ble til at scrum-master, sammen med en annen kunderepresentant administrerte produktbakloggen. Det forklares at produkteieren hele tiden er den som tok beslutningene om prioriteringer og endringer, men da med sterk veiledning fra Scrum-master.

Sprint

Teamet valgte å ta i bruk en sprintlengde på tre uker. Begrunnelsen for dette valget var at Kunden kun hadde satt av midler til å kjøre en iterasjon til denne piloten. Valget ble da tatt på tre uker, da det var det økonomiske grunnlaget.

"Det det gikk på der, var at de hadde fire uker, og noe av det gikk rett og slett på kostnad. De har bare fått penger for å kjøre en iterasjon av kunden sin. # Og da ble det litt i forhold til hvor mye penger som var tilgjengelig. At da hadde vi definert fire uker. Så hadde han kanskje steilet litt på pengene. Så det var rett og slett en økonomisk begrunnelse. Han hadde kanskje blitt mer happy med, hvis vi hadde tatt det på to uker, for da hadde det blitt enda mindre penger som hadde gått ut, men du skal kunne levere noe og der er det." (Scrum-master, prosjekt 2)

Iterasjonstiden ble derfor valgt etter lengste mulige iterasjonslengde i forhold til budsjett, slik at det ble mulig å gjennomføre en fullverdig iterasjon.

Daglige møter

Kundeorganisasjonens organisasjonskultur påvirker hvordan prosjektet følger opp møtene i prosjektet. I denne organisasjonen oppleves det som om kundens ansatte ikke tar

seriøst på faste tidspunkt, noe som skaper problemer for de daglige møtene. Det beskrives at det er normalt at prosjektdeltagerne dukker opp ca 5min etter fastsatt tid. Denne bedriftskulturen har medført at scrum-master har måtte tilpasse når de daglige møtene skal avholdes.

”Vi skal ha et fast tidspunkt hver dag. Det kan være at det blir kl ti for å garantere at folk er der. For de er; kulturen er sånn.” (Scrum-master, prosjekt 2)

Siden de daglige møtene er såpass avhengige av deltagelse fra hele scrum-teamet er det derfor om å gjøre å plassere møtet til en tid hvor samtlige deltagere er tilstede på jobb og ikke opptatt med andre gjøremål.

Sprintgjennomgang

Datagrunnlaget sier ikke så mye om prosjekt 2 sin gjennomføring av sprintgjennomgangsmøtet, annet enn at det er et av møtene som gjennomføres.

Det ble derimot foretatt observasjon av et spintgjennomgangsmøte i prosjekt 1, der scrum-master for prosjekt 2 fasiliterte møtet som en ekstern resurs i prosjektet.

Andre praksiser funnet i prosjektet

Selve prosjektet ble satt i gang etter en presentasjon for ledelsen i kundeorganisasjonen om Scrum, slik at ledelsen hos kunden støtter fullt opp om denne prosessen. Kunden har også testet ut Scrum på forhånd, uten ekstern kompetanse.

4.2.3 Prosjekt 3 – Logistikkssystem

I Prosjekt 3 har bedriften fått jobben å utvikle et større logistikkssystem i samarbeid med kunde. Bedriften deltar med scrum-master og utviklere. Prosjektet har hatt flere utfordringer i forhold til valg av metodiske prinsipper, Scrum-master er plassert på Sørlandet, mens prosjektet foregår i en annen landsdel.

Scrum-team

Tabell 7 - Prosjekt 3: Prosjektdeltagere

| Rolle | Organisatorisk tilhørighet |
|-----------------|-----------------------------------|
| Scrum-master | Bedriften |
| Produkteier | Samarbeidsbedrift |
| 4x Utviklere | Bedriften |
| 3x Kravstillere | Kunde |

Problemer med henhold til kommunikasjon om prosessen i prosjektet har ført til at scrum-master har overført mye av kommunikasjonsansvaret mellom kunden og prosjektet til kundens prosjektleder. En løsning som betegnes som nødvendig i og med at scrum-master for prosjektet ikke er geografisk plassert i nærheten av scrum-teamet og kunden.

”Kundens prosjektleder, han driver kun med politikk. Jeg har alt det administrative prosjektorganisering og sånt noe, så har han møter og ordner alt det politiske da.”(Scrum-master, prosjekt 3)

Slik utnytter scrum-master at kundens prosjektleder kjenner kundens egne normer bedre enn bedriften for å trenge gjennom til en utfordrende kunde.

Scrum-teamet i dette prosjektet har isolert seg fra kunden, ved å plassere seg i et annet bygg enn kunden. Bedriften har tidligere utviklet andre løsninger for den samme kunden, og det beskrives som et problem at representanter for kunden forstyrrer scrum-teamet med spørsmål rettet mot disse systemene. Valget med å isolere seg er derfor et bevisst valg scrum-master for prosjektet har gjort, for å hindre støy fra ”kyllinger”.

”Vi har jo flere systemer vi har utviklet for de tidligere, så mange kommer inn til de og spør spørsmål om andre ting en selve prosjektet.” (Scrum-master, prosjekt 3)

Som sitatet over nevner, har det vært en utfordring med spørsmål og henvendelser i forhold til tidligere utviklede løsninger for kunden.

Produktbaklogg

For å definere i produktbakloggen, har prosjektet valgt å benytte seg av brukerhistorier for å beskrive disse. Brukerhistoriene er laget med bakgrunn i iterasjonslengden. Produktbakloggen viser derimot at ikke alle brukerhistorier følger den malen som er satt i forhold til hva en brukerhistorie skal inneholde, da flere av brukerhistoriene ikke inneholder hvilken verdi den gir til systemet. Brukerhistoriene i produktbakloggen inneholder i tillegg til selve brukerhistorien en eventuell beskrivelse, kategorisering, om den er avhengig av noe annet og en status. Brukerhistorien er estimert i en poengskala basert på Planning Poker. Brukerhistoriene i produktbakloggen er prioritert med skalaen 1_1,1_2, 1_3, 2_1 osv.

Sprint

Scrum-master har besluttet å benytte seg av en iterasjonslengde på to uker i dette prosjektet. Bakgrunnen for det valget er at scrum-master mener det er viktig å få tilbakemeldinger fra kunden hyppigere enn ved en eventuell 30 dagers iterasjon. Scrum-master presiserer at han ikke mener dette valget er uproblematisk og at det ikke medfører større overhead i prosjektet.

Daglige møter

Distansen mellom scrum-master og scrum-team gjør det vanskelig å holde daglige møter ansikt til ansikt. De har løst utfordringen ved at Scrum-master gjennomfører det daglige møtet over telefonkonferanse. Scrum-master uttrykker at han er svært strikt på prinsippet om å ha 15minutters tidsavgrensning på dette møtet.

Sprintgjennomgang

Prosjekt 3 benytter seg av testkriterier fra kunden, som i prosjekt 1 og disse ble testet i slutten av iterasjonen sammen med en full systemtest.

Andre praksiser funnet i prosjektet

Mangelen på fleksible løsninger for prosjektstyring i organisasjonen har ført til at scrum-master for dette prosjektet har utviklet et eget prosjektstyringsverktøy spesielt tilpasset dette prosjektet. I dette verktøyet rapporterer utviklerne inn timeforbruk og arbeidsoppgaver skjematisk. Verktøyet er et Excel dokument, som inneholder en digital fremstilling av produktbakloggen, sprintbakloggen, timeforbruk pr brukerhistorie, tilgjengelige resurser og hastighet i prosjektet i forhold til poeng pr iterasjon. Samtidig jobber dette prosjektet mot en kunde som følger sitt eget system for rapportering og prosess, noe som scrum-master har måtte tilpasse seg. Det vil si at bedriftens interne prosjektstyring følger en prosess, og bedriftens eksterne rapporteringer er tilpasset en annen.

Scrum-master beskriver det som en utfordring å benytte seg av smidig utvikling ovenfor denne kunden. De representantene bedriften har kontakt med hos kunden er personer med lang fartstid i bransjen. Det antydes at kunden har sett på mulighetene i Scrum, men at det ikke har falt i smak for dem. Scrum-master presenterer dette som en mulig forklaring på hvorfor kunden motarbeider prosjektets prosess.

”Ja, altså de som jobber på it og de som styrer, har vært i militæret i x antall år og er i 50-60åra, og de er ikke så lette å snu.”(Scrum-master, prosjekt 3)

4.2.4 Prosjekt 4 – Statistikkportal

Prosjekt 4 er et prosjekt hvor bedriften skal utvikle en portal for en kunde. Denne kunden ønsker at brukerne skal kunne logge på portalen, registrere statistiske data og hente ut statistikk samlet om sitt eget, og bransjenære selskaper. Bedriften har gjennom anbudsrunder promotert bruk av smidig utvikling for prosjektet, noe som har fått kundens oppmerksomhet. Prosjektet har en lengde på ca to måneder, og er et relativt lite prosjekt.

Scrum-team

På datainnsamlingstidspunktet var ikke prosjektets deltagere fastsatt annet enn kundens produkteier og bedriftens Scrum-master.

Tabell 8 - Prosjekt 4: Prosjektdeltagere

| Rolle | Organisatorisk tilhørighet |
|--------------|----------------------------|
| Scrum-master | Bedriften |
| Produkteier | Kunde |

Scrum-master i prosjektet er samme personen som er Scrum-master i prosjekt 3. Produkteieren i dette prosjektet beskrives som ung og nyutdannet.

”Der har vi en ung mann som er produkteier. Og, han var vel ferdig å studere til sommeren tror jeg. Så han er bevandret og kan bruke en datamaskin. Han har tenkt mye på dette her og har mange ideer.”(Scrum-master, prosjekt 4)

Scrum-master uttrykker at disse egenskapene kommer til å føre til at prosessen i forhold til produkteier kommer til å gå bedre enn i andre prosjekter scrum-master har gjennomført.

Produktbaklogg

Prosjektverktøyet ble benyttet også i dette prosjektet til å administrere produktbakloggen. Dette verktøyet ble også benyttet på storskjerm under generering av brukerhistorier sammen med kunden. Disse brukerhistoriene ble generert i forhold til mal som beskrev en brukerhistorie slik: "Som <rolle> skal jeg <funksjon> fordi <verdi>". I tillegg til dette ble brukerhistoriene kategorisert og tildelt eventuelle merknader som forklarte brukerhistorien ytterligere.

Sprint

Sprintlengden i prosjektet er satt til to uker, og er forklart ved den korte lengden på prosjektet. Scrum-master forklarer det slik:

"..det er jo det også for eksempel når en nå har to måneder å lage et produkt på, da synes jeg ikke det er helt fornuftig å ha fire ukers iterasjoner heller. Da rekker jeg så vidt to iterasjoner. Og da føler jeg ikke en får den styringa underveis, som kanskje er interessant." (Scrum-master, prosjekt 3)

Behovet for å styre prosjektet, fører derfor til at det gjøres et avvik fra Scrum sine definerte 30 dager pr sprint.

Daglige møter

På datasamlingstidspunkt hadde det enda ikke blitt gjennomført noen sprinter, så det foreligger ikke data om gjennomføringen av de daglige møtene i dette prosjektet.

Sprintgjennomgang

Som i de daglige møtene, var det ikke gjennomført noen iterasjoner ved datainnsamlingstidspunkt, noe som medførte at datagrunnlag for dette punktet ikke eksisterer.

Andre praksiser funnet i prosjektet

For å bli enige med kunden om hvilke krav som stilles i forhold til å styre prosjektet etter Scrum, og hvordan prosessen kom til å være videre, ble det gjennomført et oppstartsmøte før brukerhistoriegenereringen for å gi kunden en introduksjon til Scrum.

4.3 Analyse av funn i prosjektene

I dette underkapitlet har jeg etter en kategorisering av praksisbruken også kategorisert de faktorer i hvert prosjekt som ut i fra datagrunnlaget har hatt en påvirkende effekt valg eller faktisk bruk av praksisene. For å illustrere faktorenes påvirkning ovenfor prinsippene har det blitt generert modeller for de praksiser det er en klar sammenheng mellom faktor og bruk. For å gi en oversikt over de metodiske prinsippene brukt i organisasjonen, hvor de er definert og hvordan de faktisk har tatt i bruk er det benyttet

Larsen, Päivänita, & Smolander (2007) sitt NIPO grid. I tabellen er prinsipper direkte knyttet til Scrum uthevet i *kursiv*.

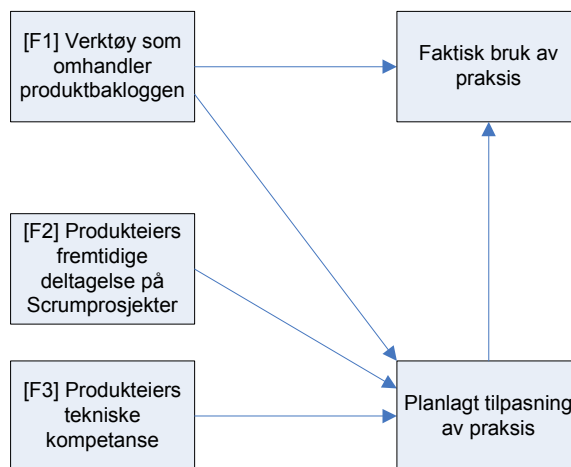
Tabell 9 - NIPO grid, organisasjonen

| Faktisk omfang av benyttet praksis | Tiltenkt omfang av definert praksis | | | |
|------------------------------------|--|-------------|---------------------------------------|---|
| | (None) Ingen | Individuell | Prosjekt | Organisasjon |
| (None) Ingen | | | | |
| Individuell | Akseptansetestkriterier benyttet som programmeringsguide | | Team-samløkalisering | |
| Prosjekt | Uferdige brukerhistorier følger automatisk til neste iterasjon | | Kodestandardisering | <i>Produkt baklogg Daglige møter Iterasjoner(Sprint) Sprintplanleggingsmøte Teamsammensetning</i> |
| Organisasjon | | | Bruk av Excel prosjektstyringsverktøy | Poengestimering |

Scrumprinsippene bedriften har tatt i bruk er lagt som grunnlag for alle smidige prosjekter organisasjonen gjennomfører og er derfor plassert under organisasjon i kolonnen for "tiltenkt omfang av definert praksis". Gjennom analyse av de forskjellige prosjektene ble det også avdekket praksiser som ikke nødvendigvis er dekket av Scrum, og det var også en differanse i hvordan de forskjellige praksissene ble benyttet i de undersøkte prosjektene.

4.3.1 Produktbaklogg

Produktbakloggen eksisterer som en liste i prosjektstyringsverktøyet i alle prosjekter, prosjektene har derimot gjort enkelte endringer på sin gjennomføring av prinsippet. I forhold til NIPO rammeverket er scrumprinsippet produktbaklogg plassert som definert av organisasjonen i form av teorien fra Scrum, men tilpasset i hvert enkelt prosjekt.



Figur 7: Faktorer som påvirker bruk av scrumprinsippet produktbaklogg

Faktor F1 - Verktøy som omhandler produktbaklogg beskriver verktøy tilknyttet prosessen og prosjektstyring i prosjektet. I flere prosjekter er et Excel skjema benyttet for å holde rede på for eksempel produktbaklogg og tilgjengelige resurser i prosjektet. Dette verktøyet kan påvirke prosjektet i den grad at dersom det kun er enkelte som har tilgang til verktøyet, overføres mye av arbeidsmengden til de personene som har denne tilgangen. Dersom det er et gap mellom hvilke verktøy en rolle har tilgang til, og hvilke oppgaver rollen i utgangspunktet var planlagt å skulle utføre, kan det oppstå endring i arbeidsoppgaver for den aktuelle rollen

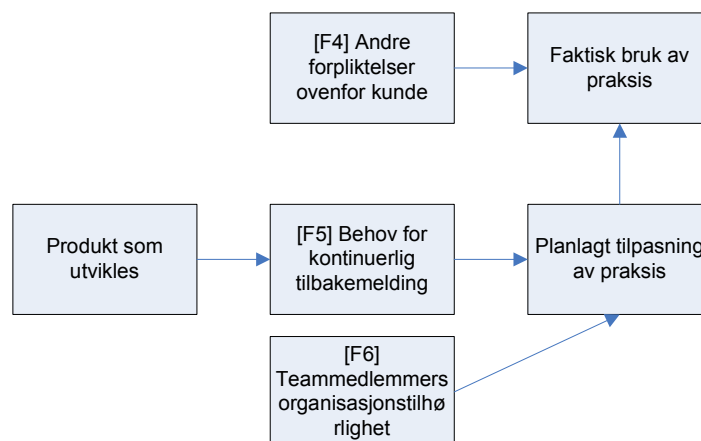
Flesteparten (Prosjekt 1, 2 og 3) av de undersøkte prosjektene har en produktbaklogg som organiseres av scrum-master og ikke produkteieren. Produktbakloggen eksisterer som en liste i prosjektstyringsverktøyet prosjektene bruker. Alle endringer og oppgaver føres inn i dette verktøyet av scrum-master gjennom prosjektets gang. Siden verktøyet inneholder god funksjonalitet til å holde rede på denne produktbakloggen, og det i hovedsak er Scrum-master som holder rede på dette verktøyet, kan det hende at bruk av et slikt verktøy vil ha en betydning for hvem som har ansvaret for produktbakloggen, dette vises også ved at den faktiske organiseringen av produktbakloggen ble utført av scrum-master i alle tilfeller. I prosjekt 4 var det planlagt å la produkteier ta fullt ansvar for produktbakloggen, men datagrunnlaget strekker seg ikke over den faktiske bruken i det prosjektet.

Faktor F2 - Produkteiers fremtidige deltagelse på scrumprosjekter beskriver om den som har rollen som produkteier skal fortsette å delta i prosjektet eller andre prosjekter senere. Denne faktoren vil mulig påvirke scrum-masters valg om å lære opp personen til å utføre de oppgaver en produkteier har eller om det er mulig å overføre disse oppgavene til en annen person. Denne faktoren har påvirket prosjekt 2 sin bruk planlagte gjennomføring av produktbakloggen.

Faktor F3 - Produkteiers kompetanse innen problemområdet kan, som i produkteiers fremtidige deltagelse på scrumprosjekter, ha en betydning for hvilke faktiske oppgaver rollen har i prosjektet. En produkteier uten kompetanse innenfor problemområdet, må mulig ha mer oppfølging og støtte fra andre, noe som kan påvirke oppgavene til rollen. Denne faktoren var gjeldende i prosjekt 2, med en produkteier som ikke hadde kompetanse eller mulighet til å kunne ta på seg det fulle ansvaret. I prosjekt 4 viste det seg derimot at produkteier hadde de rette egenskapene og det var planlagt at han skulle utføre denne rollen til det fulle.

4.3.2 Scrum-team

Scrum-team er plassert i NIPO rammeverket som definert av organisasjonen men faktisk tilpasset i forhold til hvert enkelt prosjekt. Antall deltagere i prosjektene varierer mellom prosjektene og det er kun et prosjekt som har et scrum-team på størrelse med hva Scrum anbefaler. Hvordan Scrum-teamet skal lokalisere seg er derimot ikke definert i scrum-teorien og det er her gjort tilpasninger pr prosjekt, men det er bare enkelte individer som følger praksisen som vist i prosjekt 3.



Figur 8: Faktorer som påvirker lokalisering av Scrum-teamet

Faktor F4 - Andre forpliktelser ovenfor kunde kan være andre utviklede systemer som er i drift nå og prosjektdeltagere som har flere roller. I tilfeller der det er andre systemer som er utviklet, kan dette skape merarbeid for prosjektteamet, da feilrettinger og annen supportvirksomhet kan være noe som må tas høyde for og kontrolleres samtidig som utvikling av nytt system foregår. Dette ble svært gjeldende i prosjekt 3, og medførte det drastiske tiltaket med å flytte hele scrum-teamet bort fra kundens lokaler.

Faktor F5 - Behov for tilbakemelding beskriver et prosjekts behov for kontinuerlig tilbakemelding fra kunden. Dersom systemet som skal utvikles er spesifisert i detalj i produktbaklogg, vil behovet for tilbakemelding være mindre enn dersom de spesifiserte brukerhistoriene ikke er detaljspesifisert. I denne oppgaven er det observert at et prosjekt hadde vesentlig større behov for tilbakemeldinger fra kunden enn de andre. Produktet som ble utviklet i det tilfellet var i stor grad designrelatert, og mye av behovet for tilbakemelding gikk på designrelaterte spørsmål. Behov for tilbakemelding kan derfor bli

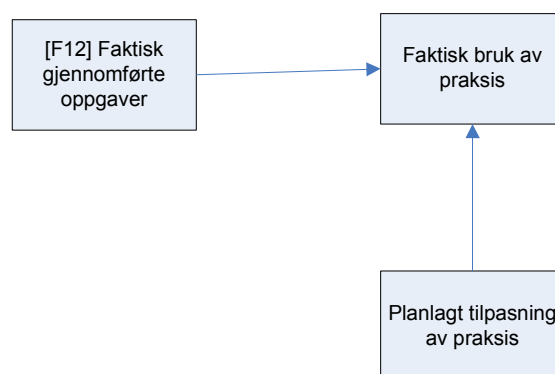
påvirket av **type system som blir utviklet** hvor produkter med stort fokus på design kan øke behovet for tilbakemelding fra kunden. I for eksempel prosjekt 1 ble det vist at behovet for tilbakemelding var relativt stort, noe som også kan ha påvirket valget om lokalisering.

Vi ser at mens det ene prosjektet eksponerer seg i større grad for kunden begrunner dette med et ønske om tilbakemelding, mens neste prosjekt prøver så godt de kan å unngå støy. En mulig årsak til dette skillet kan ligge i produktet som utvikles. Reiselivsportalen er et designprosjekt, hvor funksjonaliteten som utvikles er direkte synlig fra et brukerperspektiv, mens logistikksystemprosjektet er et funksjonalitetssystem, hvor funksjonaliteten ikke visualiseres i samme grad. I reiselivsportalprosjektet kan kunden med enkelhet kommentere fargevalg, plassering og brukskvalitet kontinuerlig, da dette ikke krever dyp teknisk forståelse.

Faktor F6 - Teammedlemmers organisasjonstilhørighet vil være om prosjektdeltagerne hører til organisasjonen som er utvikler, om de er fra kunden eller om de er fra 3.part. Tilhørigheten til forskjellige organisasjoner kan medføre forskjellig organisasjonskultur innad i prosjektet. Denne faktoren har vist seg å påvirke valget av lokalisering, i prosjekt 1 hvor det er utviklere fra flere bedrifter, har teamet plassert seg hos kunden, i prosjekt 2 er en lokalisering hos kunden også naturlig, da samtlige utviklere er fra kundens organisasjon.

4.3.3 Sprintplanleggingsmøte

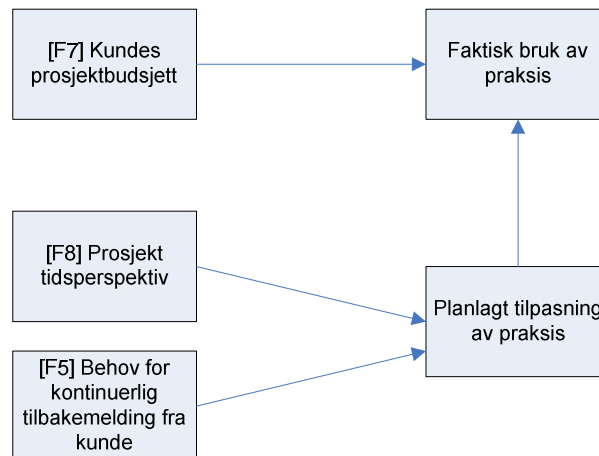
Sprintplanleggingsmøtet er definert i scrumteorien og de fleste prosjektene følger scrumteorien under disse sprintplanleggingsmøtene. Allikevel er det noen unntak, som ved at uferdige oppgaver følger automatisk med til neste iterasjon i prosjekt 1. Prinsippet plasseres derfor i NIPO rammeverket som definert av organisasjonen, men tilpasset i hvert enkelt prosjekt. Praksisen ved å videreføre brukerhistorier direkte til neste iterasjon dersom de ikke var ferdige var en såpass markant praksis i det ene prosjektet at det listes i NIPO som en individuell praksis, ikke formalisert definert, men utført av et prosjekt.



Figur 9: Faktorer som påvirker prinsippet sprintplanleggingsmøtet

Faktor F12 – Faktisk gjennomførte oppgaver har påvirket sprintplanleggingsmøtet i enkelte prosjekter ved at oppgaver som ikke er gjennomførte automatisk blir tatt med videre i neste iterasjon. Arbeidsmengden disse medfører, vil påvirke hvor stor arbeidsbelastning scrum-teamet kan ta på seg.

4.3.4 Sprint



Figur 10: Faktorer som påvirker bruk av scrumpraksissen sprint

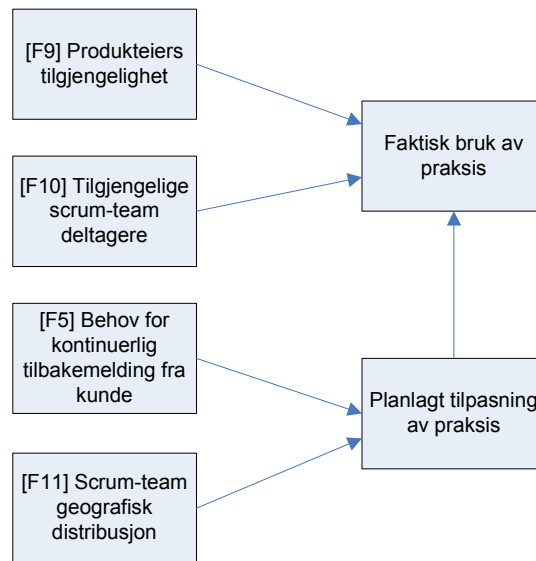
Faktor F7- Kundes prosjektbudsjett sier noe om hvilke begrensninger kunden setter til prosjektet. Dette kan påvirke både resurser, verktøy, lokalisering og lengde på prosjekt og iterasjoner. Denne faktoren hadde en begrensende virkning i prosjekt 3 der prosjektbudsjettet hindret Scrum-master å gjennomføre en full sprint på 30 dager.

Faktor F8 - Prosjekt tidsperspektiv beskriver hvilken tidsramme det er satt på prosjektet. I prosjektene som er undersøkt er det funnet klare linjer mellom prosjektets tidsperspektiv og hvor lange iterasjoner som benyttes. I korte prosjekter på noen få måneder opplever prosjektlederne at det blir for få styringspunkter underveis.

Faktor F5 – Behov for kontinuerlig tilbakemelding fra kunde kommer igjen som begrunnelse for de planlagte endringene på sprintlengde. I det prosjektet som hadde størst behov for tilbakemelding var prosjektlengden kortest.

4.3.5 Daglige møter

Prinsippet daglige møter ble benyttet av samtlige prosjekter, men det vises en forskjellig hensikt i hvorfor dette møtet ble gjennomført. Den aktive bruken av produkteieren i et av prosjektene medførte en større kommunikasjonsflyt i prosjekt 1. Praksisen er plassert i NIPO som definert av Organisasjonen, men tilpasset av prosjektene.



Figur 11: Faktorer som påvirker bruk av scrumprinsippet daglige møter

Faktor F9 - Produkteier tilgjengelighet beskriver produkteierens tilgjengelighet ovenfor scrum-teamet og prosjektet. I de tilfeller der produkteieren deltok på de daglige møtene, var produkteierens tid tilrådighet en begrensende faktor på hvor ofte produkteier deltok på de daglige møtene.

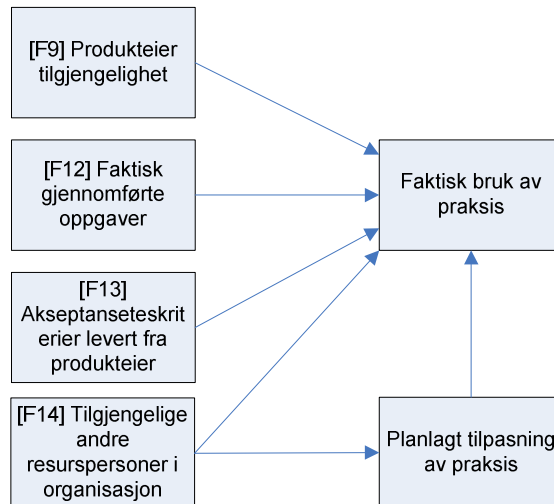
Faktor F10 - Tilgjengelige scrum-team deltagere omhandler deltagere som er tilgjengelige til å kunne delta i prosjektets oppgaver. Dersom det er slik at prosjektdeltagerne av forskjellige årsaker er bortreist, syke eller opptatt med andre gjøremål i forhold til arbeidet i organisasjonen eller i prosjektet kan dette påvirke gjennomføringen av de oppgaver prosjektets prinsipper innebærer. I de daglige møtene fikk gjennomføringen en annen form dersom enkelte scrum-team deltagere ikke var tilstede. Dersom for eksempel Scrum-master ikke var tilstede fungerte det daglige møtet som en oppdatering mellom resten av scrum-teamet. Møtets form og struktur ble også endret dersom få deltagere var til stede, hvor møtet ble mer uformelt da det var få deltagere til stede. Organisasjonskultur i bedriften kan også styre denne faktoren, som for eksempel i prosjekt 2, hvor den organisatoriske kulturen tilsa at det var greit å komme litt sent til møtene.

Faktor F11 - Scrum-team geografisk distribusjon er faktoren som beskriver hvordan scrum-teamet er lokalisert i forhold til hverandre. Dersom avstand er stor, vil det være nødvendig å benytte seg av andre kommunikasjonskanaler enn muntlig. I prosjektene ble telefonkonferanse benyttet som alternativ til å møte opp i egen person.

4.3.6 Sprintgjennomgang

Sprintgjennomgangsmøte blir gjennomført av samtlige prosjekter for å avslutte en sprint, men det er visse forskjeller på om sprinten blir avsluttet til fastsatt tid eller om det skjer en utsettelse på grunn av mangel på tid fra produkteier eller på grunn av manglende

resultater i løpet av en Sprint. Prinsippet blir derfor plassert i forhold til NIPO som definert av organisasjonen og scrumteorien, men tilpasset til hvert enkelt prosjekt.



Figur 12: Faktorer som påvirker bruk av scrumprinsippet sprintgjennomgang

Faktor F9 - Produkteier tilgjengelighet hadde en effekt på sprintgjennomgangen ved at produkteier ikke hadde tid til å gjennomføre møtet på avtalt tidspunkt hver uke i prosjekt 1.

Faktor F12 - Faktisk gjennomførte brukerhistorier er de brukerhistoriene som er utviklet i løpet av en iterasjon, denne faktoren ble funnet som en faktor for om det var nødvendig å ta en akseptansetest i løpet av en iterasjon i et av prosjektene. I noen tilfeller ble det ikke fullført nok brukerhistorier til at scrum-teamet så det nødvendig å foreta en sprintgjennomgang.

Faktor F13 - Akseptansetestkriterier levert fra produkteier beskriver om det eksisterer akseptansetestkriterier til brukerhistoriene i sprinten. Dette vil ha en betydning for hvordan sprintgjennomgangen faktisk ble gjennomført. Dersom det ikke forelå akseptanseskriterier, ble det vanskelig å teste om brukerhistorien stilte til produkteierens krav. I prosjekt 1, ble det i flere tilfeller til at akseptansetesten måtte utsettes, da produkteier ikke hadde tid til å lage testen før iterasjonsslutt.

Faktor F14 - Tilgjengelige andre resurspersoner i organisasjon beskriver om det eksisterer resurspersoner som kjenner domenet eller prosessen bedre enn prosjektteamet. I disse tilfellene kan organisasjonen benytte seg av denne kunnskapen i prosjektene og planlegge ut i fra dette. I bedriften eksisterte det en praksis på å ta inn resurspersoner fra bedriften til å veilede under sprintgjennomgangen, men dette skjedde kun dersom disse ressursene var tilgjengelige og ikke opptatt med andre prosjekter.

4.3.7 Brukerhistorier

Samtlige prosjekter oppgir at de har benyttet brukerhistorier for å forklare oppgavene i produktbakloggen. Det er derimot en varians i om alle brukerhistoriene er utformet i

forhold til beskrivelsen. Det er lignende tendenser i prosjekt 3 også, hvor et fåtall av de brukerhistoriene det er gitt tilgang på har med verdi i formateringen. I prosjekt 3 er brukerhistoriene beskrevet med større fokus på det programmeringstekniske enn i prosjekt 1.

4.3.8 Prosjektstyringsverktøy

Prosjektstyringsverktøyet ble konstruert spesielt for prosjekt 3 etter at Scrum-master fant mangler i eksisterende timeregistreringssystem og prosjektstyringsverktøy i organisasjonen. Dette prosjektstyringsverktøyet har også blitt tatt i bruk av de andre prosjektene i organisasjonen som benytter seg av Scrum og plasseres derfor i P-O i NIPO rammeverket. Scrum-master for prosjekt 1,2 og 4 har derimot kun benyttet deler av skjemaet, da de forklarer at de ikke har behov for den detaljerte sporingen skjemaet legger opp til. Samtlige prosjektledere benytter seg av verktøyets funksjonalitet for estimering av fart i prosjektet i en "burn-down" graf. Burn-down grafen har blitt en standard for hvordan en viser prosjektets status på. Disse grafene blir også benyttet i statusrapporten fra Scrum-master i prosjekt 1 til kunden.

4.3.9 Poengestimering

Samtlige prosjekter estimerer omfanget av produktbakloggen i poeng, som så kan regnes om til timetall. Allikevel antydes det at noen av de prosjektene som har foregått over en viss tid benytter seg av andre metoder for å estimere tid. De prosjektene omfattes ikke av denne oppgaven, ettersom de ikke benytter seg av Scrum som basis for utviklingsmetodikken.

"Det er godt mulig noen av de gamle gutta fremdeles estimerer i timer. Men det går ikke innom vårt område uansett da." (Scrum-master, prosjekt 1)

Siden det ikke eksisterer noe datagrunnlag til å si om andre prosjekters bruk av estimeringsteknikker annet enn at en person sier det er en mulighet for at det kan eksistere plasseres prinsippet poengestimering som definert og benyttet av hele organisasjonen i NIPO rammeverket i denne oppgaven.

For å komme frem til poengestimatet i produktbakloggen er det benyttet planning poker i alle prosjektene.

4.3.10 Kodestandardisering

Det er enkelte antydninger til at kodestandardisering er blitt benyttet i prosjektene, men denne praksisen er ikke dokumentert i noen form. Praksisen er derfor plassert som definert og benyttet i enkelte prosjekter i NIPO rammeverket.

5 Diskusjon

5.1 *Faktorer som påvirker bruk av de metodiske prinsippene i organisasjonen*

I prosjektene undersøkt i denne oppgaven ser vi at de fleste benyttede utviklingsprinsipper og teknikker har en rot i det teoretiske grunnlaget. Strukturen på prosjektene, de viktigste komponentene og prinsipper som er hentet inn er diskutert i litteratur og fagkretser i forskjellig grad. I forhold til Fitzgerald, Russo og Stolterman (2002), er dette i tråd med den formaliserte metoden som basis for metode i aksjon. Scrum-prinsipper ligger til grunn for samtlige prosjekter undersøkt og det vises også en viss form for rasjonalisering av metodens rolle. Økt kontakt og bedret kommunikasjon med kunden er bakgrunnen for hvorfor organisasjonens bruk av metodikk fremkommer slik den gjør i dag. De individuelle prinsipper benyttet i hvert enkelt prosjekt er imidlertid tilpasset på grunnlag av flere forskjellige faktorer. Studien av Baskerville & Pries-Heje (2004) viste at de undersøkte danske firmaene jobbet mye med å takle vage spesifikasjoner. Dette tilpasset de seg ved å jobbe iterativt med spesifisering og benytte seg av prototyping. Scrum løser dette ved hjelp av den "levende" produktbakloggen som hele tiden defineres av produkteier etter hvert som forståelsen for produktet blir større (Schwaber & Beedle, 2002).

5.1.1 Faktorerens påvirkning i forhold til Scrumprinsippene

Produktbaklogg

Fitzgerald et al.(2002) sin beskrivelse av at utviklingskonteksten former bruken av metodikken kan settes i direkte sammenheng med faktor F1 verktøy, som påvirker gjennomføringen og ansvarsforholdet til produktbakloggen. I forhold til dette rammeverket og Scrum, passer ikke produkteieren inn i grupperingen utviklere i "Method-in-action" rammeverket, men må om nødvendig plasseres som en del av utviklingskontekst. Kundens aktive rolle i prosjektene som produkteier passer ikke direkte inn i noen av klassifiseringene, da kunden helt klart former bruk av metodikken ved å sette grenser for tid og ressurser, men samtidig påvirker direkte utviklingen ved å være en testperson og retningsstyrer i prosjektet. Produkteiers fremtidige deltagelse vises som en betydelig faktor for hvilke oppgaver som blir gitt til rollen, det samme gjelder for kompetanse og kjennskap til domenet. I tilfeller der produkteieren ikke skulle delta på andre scrumprosjekter senere, ble han heller ikke opplært i å ta den fysiske rollen som produkteier i forhold til produktbakloggen. I stedet ble rollen utført av scrum-master og eventuelt en tredje person som skulle ta produkteierrollen i senere prosjekter. Martin et al.(2003) siterer Beck og Fowler (2001) slik:

“All the best talent and technology and process in the world will fail when the customer isn't up to scratch” (Beck & Fowler, 2001, s17)

Med sitatet over i bakhodet, kan det virke som om det er nettopp dette, Scrum-master i disse prosjektene har tatt høyde for ved disse tilpasningene som er gjort i produktbakloggen.

Scrum-team

Scrum legger ikke noen føringer på om Scrum-teamet skal ha sitt faste arbeidssted. Det som blir nevnt er at Scrum-teamet skal kommunisere gjennom de faste møtene; sprintplanlegging, daglig møte og sprintgjennomgang og at Scrum-teamet skal konsentrere seg om å bruke enkle løsninger for å kommunisere prosjektfremgang og problemer med blant annet en enkel oversiktstavle for å vise progresjonen i en sprint (Schwaber & Beedle, 2002). Alistair Cockburn (2002) vektlegger samarbeid og muntlig kommunikasjon som hovedkommunikasjonsform i prosjektene og påpeker at et samlokalisert prosjektteam skaper en bedre arena for å kommunisere.

Behovet for tilbakemelding fra kunden helt klart hatt en påvirkning i forhold til hvordan prosjektene har valgt å plassere seg. I prosjekt hvor det er et høyt ønske om tilbakemelding har scrum-teamet plassert seg i naborommet til kunden, i et prosjekt med mindre krav til design samtidig som prosjektet har andre systemer utviklet og i drift hos kunden. har scrum-teamet isolert seg fra kunderepresentanter. Selv om det er stor differanse i hvordan prosjektet er lokalisert, kan begrunnelsen for valget knyttes opp mot smidig teori i begge tilfeller. Prosjekt 3 som har valgt å isolere seg, har gjort dette for å fysisk fjerne støy fra de tidligere prosjektene som ikke omhandler den gjeldende sprinten, og slik konsentrere seg bedre om gjeldende Sprint (Schwaber & Beedle, 2002). I prosjekt 1 ble behovet for ytterligere kundecontact løst ved å samlokalisere seg med kunden i prosjekt 1 noe som er i tråd med "The agile manifesto" (Cockburn, 2002).

Det blir vist at dersom scrum-teamet har andre forpliktelser ovenfor den samme kunden enn hva som innebæres i gjeldende Scrumprosjekt, forsterkes behovet for å fjerne støy mot nåværende prosjekt. Kyllinger som forstyrrer grisenes arbeid mot sprintmålet kan være en stor utfordring å takle.

Graden av formelle regler har vært lave og i tråd med å ha så liten metodestørrelse i mindre og ikke kritiske prosjekter(Cockburn, 2000), selv om dette har vært et resultat av størrelsen på prosjektet istedenfor et bevisst valg scrum-master har gjort. Det ble sett at et distribuert prosjektteam, krever noe mer struktur enn et samlokalisert scrum-team. Distribuerte team benyttet seg av større grad av formalisert dokumentering av prosjektet for å kompensere for mindre muntlig kommunikasjon mellom scrum-master og scrum-team.

Sprintplanlegging

Sprintplanleggingsmøtet og gjennomføringen av dette prinsippet har blitt gjort relativt likt i de fleste prosjekter og i de fleste tilfeller plukkes fra produktbakloggen. Allikevel er avviket i prosjekt 1 ved at uferdige oppgaver automatisk blir tatt med videre i neste sprint en fastsatt norm i prosjektet. De faktiske resultatene i forrige sprint har derfor en direkte påvirkning til hvordan sprintplanleggingen gjennomføres i prosjekt 1. Det fremkommer ikke fra datagrunnlaget om hvordan en eventuell omprioritering av en uferdig brukerhistorie ville blitt prioritert. Scrum-master i prosjektet beskriver situasjonen som at det ikke ville blitt noen omprioritering av brukerhistoriene uansett og at valget med å ta

med brukerhistorien er gjort for enkelhets skyld, slik at en ikke trenger å fjerne en oppgave i slutten av uken for så å flytte det tilbake til rett plassering i sprintbakloggen igjen neste uke.

Sprint

Baskerville og Pries-Heje (2004) viste at tidspress var en viktig faktor i danske bedrifter med korte sykluser på iterasjonene. I deres studie ble det funnet at tidspress førte til utgivelsesorientering. Den samme studien viste at amerikanske prosjekter hadde en desperat kamp om å komme fort til markedet med sitt produkt, noe som påvirket deres metodikk til å bruke korte sykluser. Syklusene var i et av firmaene beskrevet helt nede i 2 til 15 dager. Dette samsvarer med funnene i denne oppgaven hvor kundens prosjektbudsjett og prosjektets tidsperspektiv har vist seg å være viktige faktorer i prosjektene i forhold til iterasjonslengde.

Behovet for kundetilbakemeldinger ble også funnet som en viktig faktor for hva prosjektene har valgt som iterasjonslengde, i samtlige tilfeller var det dette som var begrunnelsen for valget av prosjektets sprintlengde. I forhold til Baskerville og Pries-Heje (2004) kan behovet for tilbakemeldinger også ha bakgrunn i at kravene er vage. Dette ble vist blant annet med at kravene endret seg underveis i sprintende i form av akseptansekriterier.

Daglige møter

Kulturelle forskjeller mellom metodikk og organisasjon ble blant annet funnet av Grossman, Bergin, Leip, Merritt og Gotel (2004) noe som skapte utfordringer for å innføre enkelte metodiske prinsipper. I denne oppgaven ble det også funnet tegn til at organisasjonskultur hadde en innvirkning på dette, ved at scrum-team deltagere ikke så det så viktig å være presis i forhold til å stille på møter, noe som skapte utfordringer med tanke på å dem holde korte og tidsavgrensede.

Den geografiske distribusjonen til prosjektet har medført at det må gjøres endringer på at de daglige møtene skjer ansikt til ansikt, selv om Cockburn (2000) påpeker at det vil minke kvaliteten på kommunikasjonen, har situasjonen ført til at det nest beste alternativet tas i bruk og møtene foregår da over telefon.

I prosjekter med få deltagere vil bortfall på grunn av sykdom eller lignende ha en større effekt enn i et prosjekt med flere deltagere. I et av prosjektene undersøkt i denne oppgaven kunne en persons fravær bety at det kun ble sittende igjen en person i tillegg til Scrum-master. Gjennomføringen av daglige møter ble derfor også påvirket av frafall i relativt stor grad. Det kan sies at graden av formalitet i gjennomføringen av prinsippene har sunket med antall deltagere tilgjengelige. Cockburn (2000) presenterte at i små prosjekter burde en velge å ha få formelle regler og rutiner fra starten av, i denne oppgaven har det derimot vist seg at dersom prosjektet mister deltagere midlertidig underveis av uforutsette omstendigheter, vil de formelle reglene minke uansett.

Fitzgerald, Hartnett og Conboy (2006) viste at det var blitt utført vellykkede prosjekter med praksiser kombinert fra XP og Scrum. I deres tilfelle ble denne tilpasningen gjort

bevisst og strukturert, mens prosjektene i denne oppgaven har en vesentlig vagere definisjon på de valgene som er tatt. Allikevel ser vi at enkelte praksiser har blitt tatt med fra XP som ikke ble tatt med av Fitzgerald, Hartnett og Conboy (2006), da blant annet praksisen med tilgjengelig kunde i prosjekt 1. Kunden var ikke med hele tiden som i XP, men tilgjengelig og deltok med aktivt på de daglige møtene.

Sprintgjennomgang

Martin et al. (2003) viste at kunderepresentanten i et XP prosjekt fort fikk for mye belastning, selv om kunden hadde alle forutsetninger for å kunne klare oppgaven. Lignende erfaringer ble trukket i prosjektene og det var problemer med å få utformet akseptansekriterier i tide, noe som medførte problemer i forhold til å teste arbeid utført i en sprint i enkelte prosjekter. Produkteiers tilgjengelighet påvirket den faktiske gjennomføringen og enkelte ganger ble hele gjennomføringen av sprintgjennomgangen utsatt på grunn av mangel på tid fra produkteier. Samtidig var de faktisk gjennomførte oppgavene fra scrum-teamets en faktor som påvirket bruken av sprintgjennomgang, og det forklares at dersom det ikke var nok oppgaver ferdig en sprint, kunne gjennomgangen også bli utsatt til det var ”nok” oppgaver å teste.

Schwaber og Beedle (2002) åpner for at alle typer resurspersoner kan inkluderes i en Sprint for å bidra til å nå Sprintens overordnede mål. I denne oppgaven ble denne typen resurspersoner tatt inn for å bidra med prosess og kvalitetskontroll i sprintgjennomgangsmøtet. Denne inkluderingen av kunnskap rundt metodikk var utelukkende positivt mottatt av Scrum-teamet og bidro til bedre forståelse for Scrumprosessen.

5.1.2 Faktorerens påvirkning i forhold til andre prinsipper

Prosjektstyringsverktøy

Benyttede verktøy, kan plasseres under utviklingskontekst i method-in-action rammeverket til Fitzgerald et. al (2002). Verktøyet ble blant annet benyttet på storskjerm under generering av brukerhistorier sammen med kunderepresentantene. I prosjekt 3 hadde scrum-teamet en egen timeregistreringspraksis, og måtte levere inn en egen oversikt over timeforbruk pr brukerhistorie. I tillegg til den vanlige timeregistreringen i organisasjonen for avlønning. Det bekreftes da at bruken av dette verktøyet har hatt en formende effekt på bruken av praksiser i prosjektene. Siden produktbakloggen er administrert av Scrum-master istedenfor produkteier, bidrar det til at prosessen med å detaljere av hver instans i produktbakloggen først må gjennom scrum-master, som så redigerer det inn i prosjektstyringsverktøyet. Dette ansvarsforholdet kompliserer prosessen for produkteier å endre krav, prioriteringer eller omfang av produktbakloggen.

Den geografiske distribusjonen kan også være grunnlaget for kreasjonen av Excel prosjektstyringsverktøyet. Avstanden mellom scrum-master og scrum-team kan ha medført at det er vanskelig å holde fullstendig oversikt over prosjektet og at det kompenseres med en større grad av dokumentering av arbeidsprosessen for å ta høyde for den manglende kontakten ellers.

5.2 *Hvordan relaterer organisasjonens bruk av Scrum seg i forhold til Scrums idealer?*

I prosjektene har de fleste tilpasninger blitt gjort for å etterkomme brukerens interesser. Denne tanken er i tråd med the agile manifesto, diskutert i kapittel 2. Allikevel er det slik at Scrum skal være effektiv fordi det er helt og holdent opp til Scrum-teamet å finne den gode løsningen. Hele tankesettet bak Scrum kommer fra innovasjonsbransjen (Takeuchi & Nonaka, 1986), men utviklingen i noen av prosjektene jeg har observert minner mer om en sekvensiell oppbygging av et definert system, med vage definisjoner på forhånd. Selv om det er Scrum som ligger til grunn for tankesettet i prosjektene, ser vi at i likhet med Paivarinta et al (2007) sin konklusjon om at metodikken blir brukt som et rasjonelt utgangspunkt og ikke nødvendigvis slik metodeforfatteren hadde tenkt, og tilpasses etter hvert som målet i organisasjonen og kunder av organisasjonen endrer seg.

Samtlige prosjektene benytter seg av de samme begrepene som omhandler benyttete praksiser, men det er relativt stor forskjell på den utførelsen som blir forklart, og den utførelsen som faktisk blir gjennomført i flere tilfeller. Det er mulig å se et gap mellom hvordan de faktiske tilpasningene i prosjektene har blitt gjennomført og hva som er målet med Scrum. De tilpasninger som er gjort i forhold til sprintlengde, scrum-team lokalisering, kundeinvolvering og sprintgjennomgang har faktisk gjort så mye med prosessen, at prosjektet ikke henter ut alle de gevinstene Scrum skal gi. Det de derimot oppnådde i prosjekt 1, er å hente ut mer kommunikasjon mot kunden, slik at sluttresultatet er i tråd med hva kunden hadde sett for seg.

Bruken av planning poker i prosjektene har blitt en organisasjonsstandard for å estimere omfang av prosjektene. Denne teknikken brukes også av organisasjonen i forkant av et anbud. Det er ikke registrert noen endringer i gjennomføringen av denne teknikken i forhold til teorien.

Schwaber & Beedle (2002) bruker ikke brukerhistorier for å definere produktbaklogg eller sprintbaklogg, men utelukker heller ikke bruken av disse. Samtlige prosjekter oppgir å benytte seg av denne praksisen, men det er funnet forskjellige måter å utforme brukerhistoriene på. Det er også individuelle variasjoner av utformingen innenfor hvert enkelt prosjekt. I enkelte prosjekter har denne variansen i detaljgraden muligens skapt et behov for at akseptansetestene blir en utvidet del av brukerhistorien.

I følge "the agile manifesto" skal individer og interaksjon foretrekkes over prosesser og verktøy (Cockburn, 2002). Prosjektstyringsverktøyet benyttet i de undersøkte prosjektene er svært omfattende, dersom all funksjonalitet i det benyttes. Dersom det viser seg at prosjektdeltagerne må bruke vesentlig av sin tid til å dokumentere hva de har gjort, med hvem, og eventuelle problemer i dette verktøyet, kan mye av gevinsten ved å bruke en smidig metodikk bortfalle. Dobbeltarbeidet prosjektdeltagerne må gjennom for å oppdatere sprintbakloggen både digitalt og på oversiktstavlen benyttet i prosjektene, vil ikke bare føre til merarbeid, men også mulige feil der et av verktøyene er mer oppdatert enn det andre. For utviklernes del, fungerer ikke prosjektstyringsverktøyet til noe annet en kopi av sin egen oversiktstavle og arbeid knyttet til verktøyet kan virke unødvendig og lite smidig for Scrum-teamet dersom de må bruke omfattende med tid på å

dobbeltdokumentere status i prosjektet. Selv om det ble funnet prioritering av brukerhistoriene i produktbakloggen var denne prioriteringen vagere enn Schwaber (2002) presenterer. Istedenfor å prioritere alle oppgaver mot hverandre, var prioriteringen på en skala slik at flere brukerhistorier kunne ha samme prioritet.

I forhold til Scrum, som spesifiserer 30 dagers sprinter(Schwaber & Beedle, 2002), er det ingen av de undersøkte prosjektene som følger dette, selv om det spesifiseres at det ikke er anbefalt å endre på denne med mindre en kjenner spesielt gode grunner for det. En av prosjektenes scrum-mastere uttrykket at 30 dagers iterasjoner i utgangspunktet er alt for lang tid. Dette kan ha bakgrunn i at dette er en konsulentbedrift hvor flere av prosjektene ikke har et større tidsperspektiv enn en til tre måneder. I så små prosjekter, vil en 30 dagers iterasjon faktisk bety at en kun kan rette seg opp etter en dårlig prioritering eller etter en ikke godkjent løsning to ganger og i praksis låse prosjektet slik det bli gjort i prosjektmodeller med en fast kravspesifikasjon. På samme grunnlag ble det besluttet i 3M å gå fra 30 dager, til to ukers sprinter i deres implementering av Scrum (Moore et al., 2007).

I sprintplanleggingsmøtet er det brukerhistorien som har blitt brutt opp og gjort om til oppgaver for sprinten, mens akseptansetestkriteriene har beskrevet utvidet en noe funksjonalitet utover dette. Dette gapet mellom akseptansetestkriterienes innhold og brukerhistoriens omfang er vanskelig å finne en forklaring på ut i fra de data som foreligger, men det kan være at kundens deltagelse på sprintplanleggingsmøtene kan gjøres noe annerledes. Slik at det er et samsvar med produkteiers oppfatning av en brukerhistories omfang og scrum-teamet sin oppfatning kan nærme seg hverandre. Under sprintplanleggingsmøtet burde det derfor legges til rette for at de kravene kunden sitter med kommer godt nok frem. Schwaber(2002) presiserer at de oppgavene med høyest prioritet også skal være de som er mest detaljerte.

I mindre prosjekter, og med en produkteier med mulighet til å stille opp på de daglige møtene. Virker det som om produkteiers deltagelse i de daglige møtene kan bidra til å oppnå en bedre forståelse mellom kunde og scrum-team. Denne praksisen kan også bidra til at kunden oppnår en bedre forståelse for hvilke utfordringer scrum-teamet har, kan derfor ha forstå hvorfor prosjekthastigheten er slik den bli presentert under sprintgjennomgang. Denne situasjonen blir derfor en blanding av kontinuerlig kundetilgjengelighet i XP som beskrevet av Koskela og Abrahamsson (2004) en produkteier som i utgangspunktet kun kommuniserer gjennom bruk av produktbaklogg, sprintplanleggingsmøte og sprintgjennomgangsmøte i Scrum (Schwaber & Beedle, 2002).

5.3 Skiller Scrum seg fra smidig metodebruk generelt?

Det som skiller Scrum tradisjonell smidig tankegang er prinsippet om at scrum-teamet organiserer seg selv. Mens kunden styrer veien gjennom å identifisere hvilke utfordringer som skal løses først, er det helt og holdent opp til scrum-teamet å finne den beste og enkleste løsningen på problemet (Schwaber & Beedle, 2002). I for eksempel Extreme Programming har kunden en større rolle i selve iterasjonsarbeidet. Dette viste seg å være en utfordring i prosjekt 1, da kunden hadde veldig spesifikke tilbakemeldinger gjennom

iterasjonen, som kunne innebære detaljer som å for eksempel flytte på en visuell knapp. Denne situasjonen minner veldig mye om det som er definert som kunderepresentanten i et Extreme Programming prosjekt (Koskela & Abrahamsson, 2004). Å benytte seg av Scrum i prosjekter der brukergrensesnitt er en vesentlig del av prosjektet kan derfor vise seg vanskelig. Små oppgaver som for eksempel å flytte en knapp, er for små til å estimere og flytte tilbake i en produktbaklogg. Blir det mange nok av de små oppgavene vil en til slutt havne i en situasjon der mange små oppgaver, tar tid fra de faktisk estimerte oppgavene som inkluderes i en sprint og prinsippet med å låse en Sprint til sprintbakloggen blir ikke ivaretatt.

5.4 Bidrag til praksis

Faktorene presentert i kapittel 4 som påvirker de metodiske prinsippene kan være verdt å merke seg for praktikere i yrket. Kunnskapen om hvilke konkrete faktorer som bidrar til endringer i spesifikke praksiser innenfor smidig metodikk er noe som ikke er dokumentert i veldig stor grad i tidligere forskning og denne oppgaven kan bidra til å belyse flere problemstillinger. En stor bidragsyter til de påvirkende faktorene funnet er egenskaper funnet hos kunden. Det kan være det er utfordringer i forhold til kundens tid og resurser eller egenskaper hos kunden som gjør at praktikere må ta høyde for dette i sitt valg av praksiser i prosjektene. I forhold til Fitzgeralds "method-in-action" rammeverk (Fitzgerald et al., 2002) nevner blant annet ikke kunden som en vesentlig faktor i forhold til hva som er faktisk bruk av metoden. Kunden og utviklernes bakgrunn ligger i begrepet utviklingskontekst i dette rammeverket. Det spesielle med smidige metodikker er derimot det at kunden blir en så vesentlig stor del av prosjektet, at de blir en del av prosjektteamet. Istedenfor å stå på utsiden og gi føringer for hva som er ønskelig, er kunden en aktiv del av prosjektets hverdag. Denne situasjonen er en relativt ny situasjon som både prosjektledere og kunder må tilpasse seg dersom smidig utvikling skal være en suksess. Nå er det slik at de fleste bedrifter ikke kan velge kundene sine selv, og utfordringen blir da å tilpasse sin egen prosjektstyring og utviklingsmetodikk i forhold til den kunderepresentanten som er tilgjengelig.

Konsulentselskaper i Norge kan ha en litt annen virkelighet enn hva som beskrives i mye av litteraturen, samtidig som prosjektene er små i forhold til antall deltagere, er de også korte i forhold til tid. Det har vist seg at prosjektene i denne oppgaven ikke kunne følge Scrum sin anbefalte lengde på syklusene, da det anbefalte målet rett og slett førte til at prosjektet mistet et viktig antall styringspunkter underveis. Samtidig har det vist seg at prosjekter med for kort syklus, fort kan gli ut i en slags "flyt", hvor det er vanskelig å få tid til å gjennomføre de formalitetene Scrum fører med seg.

Usikkerhet med tanke på krav er faktor praktikere absolutt burde vurdere å sette fokus på. I Scrum skal kravene være fastsatt ved iterasjonsstart og låses slik at scrum-teamet kan jobbe mot det fastsatte målet (Schwaber & Beedle, 2002). Dersom det ikke er mulig å gjennomføre dette prinsippet, kan det være en tanke å se til metodikker som åpner for en mer åpen håndtering av krav og spesifikasjoner underveis i en iterasjon. Enkelte av prosjektene i denne undersøkelsen brukte i praksis en annen metodisk tilnærming enn hva Scrum sier, men pakket metodikken inn i et begrepssett og en struktur som er likt Scrum.

Det viste seg at det ikke var definert opp rutiner for akseptansetester i prosjekt 1, men dersom scrum-master på et tidlig stadium i prosjektet hadde implementert gode rutiner og eventuelle automatiske tester som vist mulig av Miller & Collins (2001) kunne det gjøre situasjonen betraktelig enklere å håndtere.

5.5 Implikasjoner for forskning

Som denne studien viser, har det blitt vist, er det ofte små individuelle forskjeller i bruken av en metodikk i forskjellige prosjekter. Disse forskjellene kan allikevel ha stor innvirkning på metodens rolle og hvilket mål metoden ender med å løse. I forhold til Fitzgerald (2002) er det vist at produkteieren i prosjektet har mye større påvirkning på metodebruken enn tidligere antatt, noe som kan medføre et skifte i fokus i overordnede modeller for metodetilpasning.

I forhold til NIPO rammeverket til Larsen og Paivarinta (2007) har jeg bekreftet at det kan eksistere metodiske prinsipper som kan plasseres som prosjektdefinert men tatt i bruk av en hel organisasjon, slik som bruk av prosjektstyringsverktøyet benyttet av prosjektene i denne oppgaven. Det samme gjelder bruken av akseptansekriterier som programmeringsguide, hvor praksisen ikke var definert i det hele tatt, men åpenbart benyttet som en guide scrum-teamet måtte følge i prosjekt 1. Verken plasseringen P-N eller O-P ble funnet i eksempelprosjektene presentert i artikkelen. Disse funnene kan dermed bidra til å bygge opp om dette rammeverket. Faktorene funnet i dette prosjektet kan også være med på å forklare varians i andre undersøkelser om systemutviklingsmetodikk og benyttes for å styrke forklaringen til senere forskning basert på NIPO rammeverket.

Et annet funn som kan være interessant for videre studier er koblingen mellom type produkt og behovet for kundetilbakemeldinger og bruken av smidige metodiske prinsipper. Cockburn (2000) nevner i hovedsak personer, hvor kritisk systemet er og prioritet som faktorer i rammeverket for valg av metodestørrelse. I denne oppgaven har systemets egenskaper i form av om fokuset retter seg mot brukergrensesnitt eller bedriftsfunksjonalitet også hatt en innvirkning på bruk av metodiske prinsipper.

Undersøkelser og erfaringer fra bruk av Scrum i andre organisasjoner i Norge kan være interessant å se videre på. Det kan være interessant å se om andre organisasjoner har gjort de samme tilpasningene til sin bruk, som den undersøkte i denne oppgaven. Dersom det viser seg å være en trend, vil dette bety at Scrum i sin helhet kan mangle viktige prinsipper i forhold til det norske foretningstiljøet.

5.6 Begrensinger i forhold til forskningsmetodikk

Faktorene funnet i disse prosjektene trenger ikke nødvendigvis være appliserbare i alle andre prosjekter eller organisasjoner. Da dette er funn gjort i en organisasjon, og ikke sammenlignet med annet enn aktuell litteratur. Datainnsamlingsmetodene som er valgt, har sin styrke i å beskrive situasjonen slik den er, og kunne forklare ut i fra hva som er observert. Ved å i ettertid teste påvirkningsfaktorene funnet i kapittel 4 med en kvantitativ tilnærming mot lignende bedrifter i Norge, vil det være mulig å ytterligere bygge opp om funnene gjort i denne oppgaven. Funnene som er gjort er basert på et bilde

av organisasjonen i løpet av en måned tids datainnsamling, noe som betyr at alle prosjektene har vært i forskjellige stadier og deres bruk av de metodiske prinsippene kan ha endret seg over tid. Som Jacobsen(2005) poengterer er det også en viss risiko for at de observerte prosjektmedlemmene har oppført seg annerledes da jeg har vært til stede enn hva de ville gjort uten en observatør. Det er også en viss risiko for at den virkeligheten som har blitt oppgitt i intervjuene ikke er reel. Allikevel er jeg relativt sikker på at dataene som denne oppgaven bygger på reflekterer virkeligheten. Dette fordi data fra både observasjoner, intervjuer og dokumentasjon forteller den samme historien i samtlige prosjekter. Det er mulig at gjennomføringen av analyseprosessen kunne blitt gjort på en noe bedre og ytterligere dokumentert måte, men allikevel er hovedlinjene fra Jacobsen (2005) og Strauss og Corbin (1998) fulgt. For å bygge opp under funnene gjort i denne oppgaven har jeg valgt å legge ved den delen av datagrunnlaget som ikke er taushetsbelagt til oppgaven, slik at leseren har mulighet til å spore funn tilbake til datagrunnlaget.

6 Konklusjon

Denne oppgaven har presentert en case studie basert på intervjuer, observasjoner og prosjektdokumentasjon i fire prosjekter i en mellomstor bedrift på Sørlandet. Ved å presentere den teoretiske bakgrunnen for praksiser benyttet av Scrum og generelle smidige metodiske prinsipper har jeg dannet et teoretisk grunnlag som beskriver den ideelle virkeligheten i et smidig systemutviklingsprosjekt basert på Scrum. Jeg har også presentert litteratur som beskriver hvordan bruk av systemutviklingsmetodikker blir tilpasset og formet av forskjellige faktorer i og utenfor en organisasjon. Datagrunnlaget er analysert med bakgrunn i denne litteraturgjennomgangen og resultatene viser at det eksisterer flere faktorer som ikke er nevnt i litteraturen som påvirker bedriftens bruk av metodikk.

Studien har vist at flere prosjekter i samme organisasjon i utgangspunktet følger den samme utviklingsmetodikken i teorien, men at det eksisterer del individuelle forskjeller fra prosjekt til prosjekt i gjennomføringen av de praksiser som er definert. Oppgaven har vist 14 faktorer som kan ha en påvirkning på prosjektenes gjennomføring av metodiske prinsipper. Den sterkeste trenden av faktorer stammer fra den personen som er kundens representant i prosjektene og det er avdekket at kundens rolle har en stor påvirkningskraft på prosjektgjennomføringen. Dette vises blant annet ved at produkteier påvirker bedriftens bruk av iterasjonslengde ved å sette grenser i forhold til budsjett og tid. Produkteieren har i disse prosjektene en aktiv rolle som deltager i prosjektene og det er sett at produkteiers leveranser til prosjektet kan påvirke faktisk gjennomføring ved at leveransene enten kommer for sent, eller at de avviker fra hva som er avtalt. I andre tilfeller har vi sett en rolleforskyvning fra produkteier til Scrum-master da det viste seg at produkteiers kunnskap om domenet ikke var tilfredsstillende.

Det er funnet en sammenheng i forhold til hva slags type system som utvikles og hvor stort behov scrum-teamet har for tilbakemelding fra kunden. I prosjekter med stort fokus på design å brukergrensesnitt har det blitt vist at scrum-master velger å ha kortere lengde på sprintende i prosjektet for å sørge for at det kommer stadig tilbakemeldinger fra produkteier på progresjonen i prosjektene. Bruken av prosjektstyringsverktøy har hatt en formende effekt på hvordan visse metodiske prinsipper utføres. Bruken av et slikt verktøy kan også ha hatt en effekt på rollefordeling i prosjektet, da den som håndterer prosjektdokumentet gjerne også har det formelle ansvaret for de elementer prosjektverktøyet omfatter.

Det er foretatt en sammenligning av prosjektets realitet og Scrums teoretiske bakgrunn og det viser seg at de faktorene som har påvirket prosjektet kan føre til at metodikken endrer seg såpass mye at den viker bort fra Scrum sine mål. Dette vises blant annet ved at produktbakloggen i flere av prosjektene ikke er like dynamisk og levende som Scrum viser til. Samtidig ser vi at selv om Scrum-teamene til en viss grad er selvorganiserte, har de ikke helt den samme friheten til å velge hvordan løsningen skal lages, men benytter seg av større grad av kontakt med kunden for å få bekreftelse på at valg er gjort i forhold til ønsker.

Sprintbakloggen som i følge Scrum skal låses etter sprintplanleggingsmøtet, viser seg å endre seg i løpet av en Sprint i flere av prosjektene. Disse endringene kommer som resultater av direkte samtaler med produkteier eller fra nye krav spesifisert i akseptansetestkriteriene. På bakgrunn av en mer dynamisk sprintbaklogg og få deltagere i prosjektet viser det seg også at produkteieren deltar aktivt på de daglige møtene, noe som viser en begrenset tilnærming til prinsippet tilgjengelig kunderepresentant fra Extreme Programming.

Det ble foretatt en sammenligning av Scrum mot generell smidig teori. Scrum i mindre prosjekter kan faktisk virke mindre smidig enn ønskelig. Dersom alle prinsipper i Scrum skal følges etter teorien i prosjekter med et tidsperspektiv på bare noen måneder, vil dette føre til at prosjektet fraviker fra flere av de punktene som er spesifisert som kjerneverdier i smidig systemutviklingsmetodikk.

Denne oppgaven vil kunne hjelpe praktikere innen yrket å få en oversikt over de faktorer som kan være en utfordring å løse i den virkeligheten mellomstore bedrifter i Norge har i dag. Prosjektene studert her har tilpasset seg et miljø hvor kunden står i fokus og har gjort tilpasninger som kan være nødvendige for andre bedrifter å gjøre i sin bruk av smidig systemutviklingsmetodikk. Målet med å dokumentere disse faktorene har derfor vært å bidra til kunnskapen om hvordan andre bedrifter kan tilpasse sin bruk av Scrum, eller ta høyde for de påvirkende faktorer ved å være bevisste på dem. Allikevel er det helt klart at videre forskning burde gjøres for å finne mer representative data fra flere forskjellige prosjekter. Og også følge flere prosjekter over en lengre periode.

Denne oppgaven har bidratt til å bygge opp om NIPO rammeverket, og bekreftet to nye felt i dette. Dette vil styrke dens teoretiske betydning som et verktøy for å analysere bruk av systemutviklingspraksiser i fremtiden. Videre forskning på tilpasning av Scrum og smidig systemutviklingsmetodikk er nødvendig for å danne et godt bilde dette i Norge. Og da spesielt på funnene gjort i forhold til tilpasning av metodiske praksiser i forhold til kunden. Står kunden like mye i fokus i resten av bransjen?

”Har kunden alltid rett?”

Referanser

- Baskerville, R., & Pries-Heje, J. (2004). Short cycle time systems development. *Information Systems Journal*, 14(3), 237-264.
- Beck, K., & Fowler, M. (2001). *Planning Extreme Programming*: Addison Wesley.
- Cockburn, A. (2000). Selecting a project's methodology. *Software, IEEE*, 17(4), 64-71.
- Cockburn, A. (2002). *Agile Software Development*. Boston: Pearson Education, Inc.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *Eur J Inf Syst*, 15(2), 200-213.
- Fitzgerald, B., Russo, N., & Stolterman, E. (2002). *Information Systems Development: Methods-in-Action*: McGraw-Hill Education.
- Grossman, F., Bergin, J., Leip, D., Merritt, S., & Gotel, O. (2004). *One XP experience: introducing agile (XP) software development into a culture that is willing but not ready*. Paper presented at the Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research.
- Haugen, N. C. (2006). *An empirical study of using planning poker for user story estimation*. Paper presented at the Agile Conference, 2006.
- Jacobsen, D. I. (2005). *Hvordan gjennomføre undersøkelser? Innføring i samfunnsvitenskapelig metode* (2nd ed.). Kristiansand, Norway: Høyskoleforlaget AS - Norwegian Academic Press.
- JavaBin. (2008). Java Brukerforening i Norge. Retrieved 26.05., 2008, from www.java.no
- Koskela, J., & Abrahamsson, P. (2004). On-Site Customer in an XP Project: Empirical Results from a Case Study. In *Software Process Improvement* (pp. 1-11).
- Larsen, E., Päiväranta, T., & Smolander, K. (2007). *The NIPO Grid - A Construct for Studying Systems Development Practices in Organizations*. Paper presented at the 16th European Conference on Information Systems.
- Manta, M., Antoine, F., Galliano, S., & Barras, C. (1998). Transcriber a tool for segmenting, labeling and transcribing speech Retrieved 12.05., 2008, from <http://trans.sourceforge.net/en/presentation.php>
- Martin, A., Noble, J., & Biddle, R. (2003). Being Jane Malkovich: A Look Into the World of an XP Customer. In *Extreme Programming and Agile Processes in Software Engineering* (pp. 1012-1012).
- Miller, R. W., & Collins, C. T. (2001). *Acceptance Testing*. Paper presented at the XP Universe. from <http://www.xpuniverse.com/2001/pdfs/Testing05.pdf>
- Mnkandla, E., & Dwolatzky, B. (2007). *Agile Methodologies Selection Toolbox*. Paper presented at the Proceedings of the International Conference on Software Engineering Advances.
- Moore, R., Reff, K., Graham, J., & Hackerson, B. A. H. B. (2007). *Scrum at a Fortune 500 Manufacturing Company*. Paper presented at the AGILE 2007.
- Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, 17(1), 2-26.
- Päiväranta, T., Sein, M. K., & Peltola, T. (2007). From ideals towards practice: paradigmatic mismatches and drifts in method deployment. *Information Systems Journal*.

- Schwaber, K., & Beedle, M. (2002). *Agile Software Development with Scrum*. Upper Saddle River, New Jersey: Prentice Hall.
- Strauss, A., & Corbin, J. (1998). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (2nd ed.). London/New Delhi: SAGE Publications.
- Takeuchi, H., & Nonaka, I. (1986). The new new product development game. *Harvard Business Review*, 64(1).
- Waters, K. (2008). User Stories. Retrieved 08.05., 2008, from <http://www.agilealliance.org/show/2025>
- Wells, D. (1999). User Stories. Retrieved 08.05., 2008, from <http://www.extremeprogramming.org/rules/userstories.html>

Vedlegg

1. Intervjuguide
2. Intervju med Scrum-master, prosjekt 1
3. Intervju med Systemutvikler 1, prosjekt 1
4. Intervju med Systemutvikler 2, prosjekt 1
5. Første intervju med Scrum-master, prosjekt 3
6. Andre intervju med Scrum-master, prosjekt 3
7. Intervju med Scrum-master, prosjekt 2 og 4

- Intervjuperson
 - Stilling
 - Prosjekt
 - Arbeidserfaring
- Prosjektet
 - Type prosjekt
 - Antall deltagere
 - Lokalisering
 - Tidsperspektiv
- Hvilke krav har bedriften i forhold til organisering av slike prosjekter?
- Rutiner
 - Hvaslags rutiner består et prosjekt av?
 - Spesifikasjon
 - Deltagere i prosjektet
 - Kommunikasjon
 - Fremgang
 - Utfordringer
 - Leveranser
 - Hva leveres
 - Hvor hyppig
 - Kvalitetskontroll
- Justering av beskrevne praksiser underveis i prosjektet
 - Hvorfor er dette endret?
 - Resultat av endringen
- Ble forrige prosjekt organisert på akkurat samme måte?

Intervjuer: Hva slags krav er det som blir stilt til deg i forhold til rapportering til dine overordnede?

Scrum-master: Internt? Jeg leverer ikke noen sånne formelle rapporter, men jeg har statusmøter med min nærmeste overordnede og så er han alltid CC på mine prosjekter.

Intervjuer: #1 CC hva er det for noe? #

Scrum-master: #2 Eller "Carbon Copy" da, altså, han får alle mailene som jeg sender ut på det. Så han vet jo nøyaktig prosjektstatus, men vi har ikke gidde å formalisere noen andre rapporter, for da blir det nok en rapport å skrive. #

Intervjuer: #1 Hvor ofte har du de statusmøtene? #

Scrum-master: #2 Vi har det en gang i uka, minst. Men det kan jo godt være at han er i Oslo og jeg er i Sverige og sånt noe, at det går lengre tid, men det er en gang i uka og vi har dessverre ikke klart å finne en fast dag, for det er helt umulig. Men det er en til to timer en dag i uka. #

Intervjuer: #1 Du sa du hadde mange rapporter å skrive, skriver du mange rapporter? #

Scrum-master: #2 Både ja og nei, jeg har jo prosjektrapportene, de rapporteres jo en gang i uka. Og nå er det egentlig bare dette prosjektet som jeg har skrive prosjektrapporter på, for det er det som er formalisert i smidigprosjektene #

Intervjuer: #1 Av de vi holder på med, vi kjører de jo og på de andre, men det blir sånn smått. #

Scrum-master: #2 Den prosjektrapporten, er det det samme som det excelskjemaet? #

Nei, vi bruker excelskjemaet for å spore fremgangen i prosjektene. Excelskjemaet er ofte mer oppdatert enn de rapportene. Rapportene inneholder typisk informasjon om timeforbruk og endringer og sånn. Det gjør ikke Excel arket. Vi har vel mulighet for issuetracking i excelarket, men jeg synes det ikke er bra på det, så jeg bruker ikke det sånn. Excel arket er greit på noen ting, men det er ikke så veldig greit på andre ting. Så selv om du tar det som et utgangspunkt, så vil alle prosjektlederne her finne sin måte å gjøre ting på.

Intervjuer: #1 Hvem er det som leser de prosjektrapportene i ettertid? #

Scrum-master: #2 Det er kunden. Eller prosjekteier. # Alt av materialet legger vi opp på prosjekttrommet, det er tilgjengelig for alle. Så det er ikke noe hemmelig her. Hvem som helst kan laste ned og se hva som skjer. For vi prøver å ha en veldig transparent prosjektgjennomkjøring. Med at det i alle fall ikke er noen skjulte agendaer inne i bildet her.

Intervjuer: Har du noen krav i forhold til når du skal estimere tiden du bruker eller prosjektet kommer til å bruke fra bedriften? Hvis du får et prosjekt og sjefen din spør hvor mye tid du vil bruke på det, hvordan vil han ha svar på det?

Scrum-master: Hvis vi har gitt tilbud, så har vi gjerne estimert det selv. Og da kommer jo gjerne estimatet i kroner egentlig. For vi lager alltid en kjapp brukerhistorier på ting, så bryter vi de brukerhistoriene ned i poeng og så time-estimerer vi et poeng, så vil vi jo ta veiledende timepris og sette for det. Og da får du egentlig hele gangen opp.

Intervjuer: #1 Så dere gjør dette allerede før dere har pratet med kunden? #

Scrum-master: #2 Det kommer jo litt an på prosjektet, hvis prosjektet er godt estimert og de vil bare ha tilbud, eller det er godt beskrevet, så kan vi jo gjøre det sånn. I de tilfellene hvor kunden sier at de vil bare ha et program og lurer på hvor mye det koster, så må vi spesifisere litt mer hva vi egentlig skal lage. # Så det kan godt være at vi ikke kan gi fastpris eller en estimert pris. Vi må si noe om investeringsrammer for det.

Intervjuer: Ja, det er jo noen ganger vanskelig å estimere pris på forhånd.

Scrum-master: Altså svaret på det spørsmålet, at det kommer an på hvor godt grunnlag det er. Hvis det er godt, så kan vi jo estimere veldig fort, hvis det er dårlig, så estimerer vi etter vi har fått det.

Intervjuer: #1 Men dere får ikke noen føringer? #

Scrum-master: #2 Tenker du på teknikker og sånn? Vi estimerer alltid i poeng # Og poengene time-estimerer vi også og ganger det med kroner. Alle gjør sånn, og alle skal gjøre sånn. Så det er ingen annen måte å gjøre det i denne bedriften på.

Intervjuer: #1 Hvordan finner dere ut hvor mange poeng en brukerhistorie har? #

Scrum-master: #2 Vi kjører jo releasplanleggingsmøte, eller det er jo egentlig ikke et releasplanleggingsmøte, det blir egentlig et estimeringsmøte. Med brukerhistorier # Så da prøver vi å definere opp hvilke brukerhistorier som gjelder, og poengsetter det. Men det er egentlig en ganske tung jobb, så du vil ikke ta det på det nivået nødvendigvis i tilbudsfasen, det vil du jo gjerne se opp mot andre tilbud du har gitt. Hva er det egentlig som ligner og avviker, så vil du komme fram til ett eller annet. Og det kan gjerne være på brukerhistoriene hvor du sammenligner de. "Hva er det den skal? Og hva er det den skal? OK"

Intervjuer: Var det sånn dere gjorde det i dette prosjektet? Hvordan gjorde dere det der?

Scrum-master: Da kjørte vi estimeringsmøte med kunden, så kunden satt her i to hele dager og så bygde vi opp en backlog med brukerhistorier, så estimerte vi hver enkelt brukerhistorier, fant timetallet og fant ut hvor lang tid dette kom til å ta.

Intervjuer: #1 Men da estimerte dere ikke i poeng? #

Scrum-master: #2 Jo da er det poengestimering. Så det er ikke noen som estimerer i noe annet enn poeng her. # I alle fall ikke av oss. Det er godt mulig noen av de gamle gutta fremdeles estimerer i timer. Men det går ikke innom vårt område uansett da.

Intervjuer: #1 Det er noen som bruker det som heter Planning Poker, brukte dere det?

Kan du forklare hvordan dere brukte det i <prosjekt 1>? #

Scrum-master: #2 Ja, vi har jo lagd våre egne stokker. Så vi kjører poeng fra en til 21. Vi stopper egentlig på 21. # Og da har vi egentlig sagt at poeng over 21 er såpass ugreit å estimere, egentlig blir det over 13, hvis det blir 21, så må vi ta en revurdering om vi skal skrive om brukerhistorien. Men, i alle fall opp til 21, og måten vi fasiliterer estimeringa, det er jo at en leser opp brukerhistorien, det er gjerne han som er fasilitator på møtet. Og så forklarer vi litt hvilken beskrivelse som er lagt på hver brukerhistorie, så blir det en runde rundt bordet om å avklare ting. Så sier en "en to tre", så stemmer alle. Så ved avvik, så må høyeste og laveste forklare seg. Og så blir det gjerne en ny runde rundt bordet. Hvor de sier: "det var ikke sånn, for det er gjort i den andre" Så blir det ny stemming. Hvis det er sånn rimelig konsensus rundt dette her, så kan en ta snittet. Eller det vil si å velge middeltallet. Det ble gjort noen ganger, for det er noen som konsekvent byr for lite. og det er noen som konsekvent byr for mye. Og det ser du. Sånn i teorien så skal du by helt til alle er enige, men det tar for lang tid.

Intervjuer: #1 Hvem er det som byr, er det utviklerne? #

Scrum-master: #2 Alle, vi samler en stor gjeng ved bordet. Og det som er veldig nyttig med det er at alle får kjennskap til produktet og hva du skal utvikle. # Designeren får høre hvordan utvikleren tenker, og omvendt. Så det er en veldig god investering i prosjektet å kjøre det sånn.

Intervjuer: Var det noen spesielle utfordringer dere hadde i dette prosjektet, hadde kunden full forståelse for alle brukerhistoriene for eksempel?

Scrum-master: Ja, det var jo det kunden skulle ha. Så det er jo greit. Kunden er med å styrer ganske mye her. Så, ikke bare dette prosjektet, men litt generelt da, så er det jo en sånn, når du driver og estimerer, så estimerer kunden ofte litt lite i forhold til alle andre. Det er veldig greit å ha kunden med, for da synliggjør du jobben med det, og så er det veldig mange kunder som har en tendens til å by lite, fordi de forbinder et høyt poeng med en høy kostnad. Og det er for så vidt sant, men når en estimerer, skal en kun estimere omfang. Det nytter ikke å lyve på det, for vi går på "time and material" alltid. Så selv om vi estimerer lite i omfang, så går vi over. Og kunden må jo betale for det okke som.

Intervjuer: #1 Var det samme kunden som var med på dette møte som sitter sammen dere på prosjektet nå eller? #

Scrum-master: #2 mhm. #

Intervjuer: Du som scrum-master, hva slags rolle og arbeidsoppgaver er det du har?

Scrum-master: Jeg kaller meg prosjektsekretær, så jeg henter kaffe til de som sitter og utvikler, og prøver å være blid og hyggelig og sørge for at de har det greit. Så skriver jeg noen, jeg skriver en rapport i uka. Om ukesprogresjonen, og ellers så kjører jeg jo iterasjonsplanleggingsmøtene, standupmøtene, retrospective møtene og demo. Vi har valgt å gjøre det litt annerledes på demo her enn hva vanlig scrum-metodologi sier. Vi kjører demo under akseptansetesten. Så når kunden får se ferdig produkt på demo, så kjører vi også akseptansetest.

Intervjuer: Har du lyst til å beskrive hvordan dere gjennomfører den akseptansetesten og kjører demoen?

Scrum-master: På iterasjonsplanleggingsmøtet så fordeler vi oppgaver, og på alle de oppgavene som blir fordelt så har kunden ansvar med å lage akseptanskriterier. Så, når hun har levert de, så begynner utviklerne å utvikle det de skal på bakgrunn av det.

Intervjuer: Så hun skal levere de før de begynner å utvikle?

Scrum-master: Ja, det skjer gjerne parallelt, må være effektiv her. Men det er en fordel at de kommer før de er helt ferdige, sånn at en ikke får masse blå bugslapper på tavla men, så utvikler i alle fall det på bakgrunn av akseptansetestkriteriene. Og så er det jo egentlig sånn at de som kjører ukesiterasjoner, så skal det kjøres på fredag ettermiddag Det er et utrolig dårlig tidspunkt å kjøre demo på og akseptansetest, så kunden er også mye borte, vi kan ikke låse oss til den dagen. Så det har gjerne vært sånn at av og til så får vi ikke testet noe den uka, det blir tatt neste uke. Eller kanskje utelatt ennå en uke, da begynner det å bli litt sånn derre kjipt. Fordi du får masse lapper på tavla da, det er mye å gå gjennom når du først går gjennom det. Så, men helt vanligvis, så vil kunden komme inn på fredagen og så har hun med seg akseptanskriteriene, så har vi en av utviklerne som kjører en demo. Og det er egentlig, bare å vise hvordan funksjonaliteten er tiltenkt. "For å abonnere på det nyhetsbrevet der, så må du bare legge inn e-postadressen inn der, og trykk på den" Så er det ferdig.

Intervjuer: Hva er forskjellen på en brukerhistorie og en akseptansetest?

Scrum-master: Brukerhistorien skal egentlig bare funksjonelt si hva som skal tilfredstilles for å oppnå en verdi. Ikke hvordan den verdien skal oppnås. Så du vil ikke ha verdibegrepet i en akseptansetest. Men du vil si hvordan en brukerhistorie egentlig skal oppfylles. Så det betyr, altså det er mange måter å skrive akseptansetestkriterier,

<prosjekt 1> har valgt en måte. Ja altså, de er ganske grunne for å si det sånn, du kan jo lage. De er jo bare en beskrivende tekst som sier hvordan kunden har lyst til å ha det når hun har tenkt gjennom det. Så har hun også vært og snakket med designerne og litt sånn. Og det kommer også veldig mye frem på de demoene, hvordan "Ja kan vi ikke gjøre det sånn?" "OK, så er det rosa lapp"; endring. Men, du kan jo kjøre det mer rigid også, med sånn punktliste. "legg inn e-postadresse" og så videre og så videre. Og da vil du gjerne også kjøre med et fast datasett, ganske enkelt for da kan du enkelt vite hva resultatet skal være. Hvis du kjører med testdata som ikke er faste, så kan du ikke automatisk sjekke hva som kommer ut.

Intervjuer: #1 Men det gjorde ikke dere i det prosjektet her? #

Scrum-master: #2 Nei #

Intervjuer: Jeg har noen eksempler som jeg kiket på, for jeg pratet litt med noen av utviklerne og det var noen av akseptanskriteriene som kom litt sent noen ganger, og at utviklerne følte at de programmerte litt i blinde. Når de kom til akseptansetest så var det masse nye oppgaver. Nye oppgaver, kan det komme inn gjennom akseptansetesten? I så fall, hva skjer da?

Scrum-master: Da lager vi, vi henger opp en gul lapp på tavla, fordi vi har bare gule lapper der.

Intervjuer: #1 Det er brukerhistorier ikke sant? #

Scrum-master: #2 Nei, eller det vil si vi har. Vi har to typer gule lapper. Det vil si en vanlig post-it og en litt større med linjer til brukerhistorier. # Så alle de post-itene, de gule vanlige, er oppgaver. I utgangspunktet så vil vi ikke ha nye oppgaver. Jeg er ikke noe glad i det. De går opp på en rosa lapp på tavla. Og så er det en bug for eksempel, skal den opp på en blå lapp. Men selvfølgelig så kan det komme veldig mye oppgaver i tillegg da, men det betyr at brukerhistorien ikke kan leveres. Så får du syke rapporter som sier at en ikke har blitt ferdig med mer enn en brukerhistorie. Så ser det ut som om farten har falt dramatisk. Så derfor prøver vi å fokusere veldig mye på at vi skal levere år vi skal levere. Men det har jo vist seg som en utfordring for <prosjekt 1>, for det er så mye å gjøre der. Og det har egentlig gjort at vi måtte bare tilpasse oss.

Intervjuer: Hva skjer i slutten av en iterasjon og på fredagen, og så er de oppgavene som ikke blir godkjent hva skjer med de?

Scrum-master: I praksis, så det går jo egentlig på en brukerhistorie da. Så hvis ikke alle oppgavene til en brukerhistorie er tatt, så går brukerhistorien tilbake i prosjekt backlog. Og så må den eventuelt på mandagen, da prioriteres opp igjen, for å ta den siste oppgaven. Det er ikke alltid vi har gjort det like rigid. Det vil si, hvis vi vet vi skal ta en brukerhistorie og gjøre den ferdig, så gidder ikke jeg å ta den ned fra tavla på fredag ettermiddag, og henge den opp igjen på mandags morgen. Fordi vi har egentlig bare latt de henge, de som nesten er ferdige, men vi kan ikke rapportere de ferdige da. #1 Så det blir ikke, i burn-down chartet så går ikke de inn. #

Intervjuer: #2 Er det noen iterasjoner hvor dere kun har jobbet med brukerhistorier fra en tidligere iterasjon? #

Scrum-master: Nei, det er alltid fler nye brukerhistorier.

Intervjuer: #1 Hvor mange brukerhistorier tar dere inn i løpet av en iterasjon? #

Scrum-master: #2 Det kommer litt an på, småhistorier de tar vi jo alltid inni mellom. Jeg vil si vi ligger vel på mellom fem og ti. #

Intervjuer: #1 Og i poeng i fart, hva ligger dere på der? #

Scrum-master: #2 20 til 25 poeng sånn snittmessig. #

Intervjuer: #1 Dere valgte en ukes iterasjonstid, hvorfor det? #

Scrum-master: #2 Fordi en ukes iterasjonstid gir kunden større mulighet til å påvirke prosessen. For i Scrum så er det sånn at kunden kan påvirke i iterasjonsplanleggingsmøtene. # Og jo kortere tidsfrist du har, jo kortere iterasjoner pleier du å ha. Skal du levere noe om 3 år, så kjører du månedsiterasjoner, for da blir det mer effektivt. Men, det er klart når du kjører ukeseiterasjoner, så går jo en halv mandag vekk i brukerhistorier og en halv fredag vekk i testing og så videre. For det er litt for stor overhead. Man mister for mye produksjonstid. Så, vi hadde nok, sånn utviklingsmessig på dette prosjektet, så vet jeg ikke hvor mye det hadde å si. Akseptansetestene ble så og si aldri kjørt på fredag, for kunden hadde ikke tid. Så vi klarte å være veldig effektive for det da. Men, grunnen til å kjøre ukeseiterasjoner er som sagt for at prosjekteier skal ha bedre kontroll på prosjektet og komme med innspill.

Intervjuer: #1 Var det et ønske fra prosjekteier? #

Scrum-master: #2 Nei det var egentlig forslag fra vår side. Prosjekteier hadde ikke noe forhold til det. Men personlig så synes jeg det er litt for kort. Du bruker så mye tid på møter. Så jeg vil helst ha minimum to ukers iterasjoner. Og Scrum anbefalingen er jo mellom to og fire ukers iterasjoner. #

Intervjuer: Har den korte iterasjonstiden som medfører overhead medført at dere har kuttet ned på de andre møtene også?

Scrum-master: Ja, egentlig. Sånn i teorien så skal kortere iterasjonstid også medføre kortere møter. Men det er jo ikke lineært dette her. Så jeg vil jo si at når vi ikke skal sette opp så mange brukerhistorier, så rekker vi.. Vi rekker standupmøtet på mandag og iterasjonsplanlegginga på en time. Det er noen ganger det har gått over, men det er fordi det har vært mye diskusjon rundt oppgavene. Så dessverre, så har det gått over. Men vi prøver å timeboxe det ganske godt. Men jeg ser om vi hadde hatt to, tre, fire ukers iterasjoner, så kunne vi glatt brukt en halv dag på det.

Intervjuer: Tilbake til det med akseptanskriterier. Vi hadde et retrospectivemøte hvor det kom et krav om at akseptanskriteriene skulle komme innen tirsdagen, var det ikke sånn? Hadde dere noen retningslinjer på hvordan de akseptanskriteriene skulle komme før dette møtet?

Scrum-master: Nei, egentlig ikke.

Intervjuer: I ettertid da, har det blitt fulgt, eller er det et problem?

Scrum-master: Nei, det er en utfordring. Det som jeg vet det handler litt om, det er synliggjøring for kunden. Hvis du sier at du vil ha akseptanskriteriene på tirsdag, kan kunden si at hun ikke har tid. Kommer det på torsdag, så sier du at det betyr: "blabla bla" Da må jeg vurdere hva jeg kan gjøre med det. Og hvis jeg ikke kan gjøre noe med det, så må jeg akseptere det. Og det er egentlig det som det handler litt om. Det kan være irriterende for utviklerne å ikke sitte på akseptansetestkriteriene, men vi må fremdeles også huske at det er kunden som betaler lønna vår. For vi tilpasser oss selvfølgelig kunden. Kunden må bare være inneforstått hva det betyr for oss. De forskjellige tingene.

Intervjuer: Dere har ikke gjort noen andre aksjoner annet enn å si at dere ønsker å ha det på tirsdagen?

Scrum-master: Nei, for å si det sånn. Dette er et veldig uformelt prosjekt til å være Scrum. Pleier å være ganske mye mer rigid styrt. Og da kan det godt være at neste gang for å få det til å funke bedre at vi sier at det er et krav at de skal være inne på tirsdag.

Hvis det ikke er inne på tirsdag, så begynner vi ikke på de. Så kan kunden velge. Men på en annen side, så er det kunden som styrer dette. Så hvis kunden sier at en skal drite i det og godkjenner det, hva skal en gjøre da?

Scrum-master: Jeg har ikke noe godt svar. For meg så er det forskjellig fra prosjekt til prosjekt. Men generelt sett, så vil jeg si at jeg gjør sånn som kunden sier og så får vi løse de utfordringene.

Intervjuer: Har det vært tenkt på å sette akseptansekriterier inn på et tidligere tidspunkt? For eksempel når en genererer brukerhistoriene?

Scrum-master: Nei, det tror jeg ikke vi kommer til å gjøre. For da bryter vi litt av scrumtankegangen. Så for oss er det veldig bevisst at det vi begynner med, det er brukerhistoriene og prioritering av de. Vi skal ikke gjøre noe mer med de før vi begynner på en brukerhistorie. Og da kan vi begynne ta test. Og det er også for de vi velger å ikke implementere de har vi ikke brukt noe tid på heller. Vi har heller bare konstatert at her er det et behov og det behovet har en verdi, men av en eller annen grunn så ble den verdien for lav nå til å implementere.

Intervjuer: Daglig møte, det har dere hatt hver dag?

Scrum-master: Så og så, men hvis folk har vært ute eller det er bare en på kontoret så gidder jeg ikke ta det. Hvis jeg og en til er der, da sitter vi.

Intervjuer: #1 Hvem andre er det som pleier å delta på det møtet? #

Scrum-master: #2 Alle som er griser deltar på det møtet. Jeg og prosjekteier, hun har gjerne noen oppgaver også. Det er ikke alltid hun deltar, men det er ikke nødvendig at hun deltar. Men hun kan godt delta for å høre litt, for da kan vi pese litt på akseptansekriterier og sånn. # Men det er egentlig meg og utviklerne. Ja det var to utviklere og en HTML designer som også var på de møtene. Og folk må gjerne komme som kyllinger, men kyllinger trenger ikke komme og grisene må komme.

Intervjuer: #1 Pleier det å være andre som har vært med, jeg var med når din sjef var med. #

Scrum-master: #2 Nei egentlig ikke. På demoen er det heller ingen som har noen formening. Vi kjører bare med prosjekteier alene. #

Intervjuer: #1 Så hadde vi retrospektive møte. Hvorfor har vi det? #

Scrum-master: #2 Retrospektivemøte har vi for å gjennomgå uka og identifisere de tingene som har vært veldig bra som vi må videreføre. Og de tingene som har vært dårlige, eller ikke dårlige men forbedringspotensialene og håndtere de. # Og det går jo på alt mulig. Hvis en av utviklerne synes en annen har vært en drittsekk, så kommer det frem i retrospektivemøtet. For da kan det være en sak som jeg kan håndtere. Så det er jo litt sånn. Vi må gå gjennom uka og finne frem det vi kan forbedre oss på det som har vært veldig bra og som vi kan ta med oss videre.

Intervjuer: #1 Hvem er det som pleier å møte på det? Er det bare dere der og? #

Scrum-master: #2 Ja, det er egentlig alle grisene. Kunden som er sånn halv gris, grisekylling, ho er med sånn av og til. Jeg synes det er greit å kjøre retrospektive kun med teamet, jeg vil helst ikke ha noen kyllinger med på de møtene. # For da blir det til at du binder deg litt til hva du sier. For det er klart å si at: "Akseptansetestkriteriene er alt for seine!" Hvis kunden sitter der.

Intervjuer: Hvordan foregår et retrospektivemøte?

Scrum-master: Det er egentlig til en fast tid da, så er det sånn: "Nå er det retrospektivemøte" og så går vi litt, igjen dette prosjektet er veldig uformelt, så sånn som

jeg pleier å gjøre det så pleier jeg å spørre om hva som har gått bra denne uka. Og det er jo ingen som sier noe på det. Jeg lurer litt på om noe går bra. (latter) Jeg vet om mye som de kunne sagt, som de ikke sier. Jeg tror det er litt mer fordi de er opptatt av det som går dårlig, og ikke går så bra. Det er ikke veldig mye, men det kan være for eksempel tilgang til servere her nede som ikke er bra og sånn. Og det er også sånne ting en kan ta opp sånn underveis, men det er veldig greit å ha et formelt møte hvor en faktisk standardiserer og sier: "Sånn skal det være, sånn skal det være og sånn skal det være". Og når en har satt opp de tingene som er bra eller som kan forbedres, så setter man opp aksjonspunkter for det. En skal ikke gå ut fra det møtet med en issue uten en løsning på det. Og møtet skal helst være timeboxet på ikke mer enn en time. Så det er, en får snakke fort. Hver iterasjon så skal det være et sånt møte.

Intervjuer: En annen konsulent fra bedriften var med på et slikt møte, har han vært med flere ganger?

Scrum-master: Det var egentlig meninga at han skulle være med på flere møter, men nå i konsulentbransjen så er det sånn at ting endrer seg. Så han måtte være Scrum-master på et annet prosjekt.

Intervjuer: Har dere gjort noen andre endringer på prosessen underveis? Startet dere på en måte som dere ikke gjør nå?

Scrum-master: Jo for så vidt... Jeg lyver jo litt hvis jeg sier at retrospective har vært på fredager. For retrospective har gjerne vært på fredag eller mandag. Mandag etter iterasjonsplanlegginga. For å gjøre det litt mer kompakt. Det har vi jo endret på vi hadde jo tenkt å kjøre det på fredag. Og akseptansetester de har jo vært veldig endret på, for de er kjørt når kunden har tid. Men det er jo bare sånne forutsetninger som en er nødt til å ta, eller forholde oss til. Vi kan jo ikke gjøre noe annet. Nei det eneste jeg kan si det er at den smidigprosessen som er kjørt på dette prosjektet, den er ekstremt smidig og egentlig veldig lite formell. Det går an å kjøre veldig mye mer formelle prosjekter enn dette. Og litt av det er også fordi at kunden har hatt veldig dårlig tid, og vi har veldig mye å gjøre. Og da blir det ikke sånn stemoderlig behandlet, men du har et gitt antall timer som du skal bruke på dine ting. Det gjelder både meg, utviklerne og kunden. Det som er veldig greit med smidig, det er at prosessen styrer seg selv. Jeg trenger faktisk ikke være der hver eneste dag. Så, det vil jeg jo trekke frem som et av de store plussene med smidig, at prosjektet organiserer seg selv og faktisk klarer seg. Jeg trenger ikke sove dårlig om natta fordi jeg ikke har klart å følge opp en sak. Det står jo på oppgaven, nøyaktig hva de skal gjøre. Så det er veldig pre for metodikken i alle fall at det er sånn.

Intervjuer: #1 Et av prinsippene i Scrum er jo Self organizing teams, og det fungerer vel bra. #

Scrum-master: #2 Ja over all forventning. De utviklerne som har vært på det, de sier det er så greit. For de slipper å ha en idiot av en prosjektleder som henger over og forteller hva de skal gjøre. Nå kan de finne det ut selv. # Og det er sånn anarki under ansvar, og det er kjempegrei

Intervjuer: Er det sånn at folk har.. Det virket som om det var et introduksjonskurs, det møtet, til Scrum, er det slik at det er n ytt for alle sammen?

Systemutvikler 1: Det vet jeg faktisk ikke, jeg har jo ikke vært med i bedriften så lenge, så det er vanskelig å si. Da jeg begynte her på Prosjekt 1 så var det jo nytt. jeg hadde jo lest noe om Scrum og hatt litt sånn teori om det på skolen, men altså hvordan det ble gjennomført i Praksis, det var jo nytt.

Intervjuer: #1 Det virker som om det er noe som har blitt innført nå nylig. #

Systemutvikler 1: #2 Det kan godt være det, jeg vet at de har hatt mange prosjekter tidligere som har kjørt Scrum. # Det er mulig at de har en liten sånn offensiv, for å få det gjennomgående, så alle vet hva det dreier seg om.

Intervjuer: #1 Kan du fortelle litt om deg selv, navn stilling og prosjektet du er i nå. #

Systemutvikler 1: #2 Systemutvikler 1 sitt navn, er vel utvikler som er min stilling. Nå jobber jeg for Prosjekt 1 på CMS prosjektet hos dem. #

Intervjuer: #1 Hva slags prosjekt er dette? #

Systemutvikler 1: #2 Det er <prosjekt internettadresse>, vi lager en reiselivsportal, promotering av ting å finne i området. Med aktiviteter og overnattinger osv. For å få solgt mer reiselivsting her nede. #

Intervjuer: #1 Hvor mange er det som jobber med det prosjektet her nede nå? #

Systemutvikler 1: #2 Vi er tre stykker som jobber med det, tre griser om du vil. Vi har en som er designer og HTML, så er det meg og en til som driver med utvikling, så har vi jo prosjektleder og den regla der holdt jeg på og si. #

Intervjuer: #1 Har dere fire fra Bedriften da eller? #

Systemutvikler 1: #2 Nei, det er meg fra Bedriften, og <Navn> som prosjektleder. Og han designeren og han andre utvikleren, de er fra <Samarbeidsbedrift>. #

Intervjuer: #1 Er det et stort firma? #

Systemutvikler 1: #2 Jeg vet ikke hvor store de er, men de er ganske mange tror jeg, de er jo et reklamebyrå. De står for det grafiske. #

Intervjuer: #1 Og du er da helt fersk? #

Systemutvikler 1: #2 Ja, så godt som. Nettopp ferdig med prøvetiden. #

Intervjuer: #1 har du vært med på noen andre prosjekter tidligere? #

Systemutvikler 1: #2 Ja, jeg begynte på et prosjekt for <kunde>, <fjernet grunnet identifisering> og det var faktureringsystemet deres vi jobbet med.

Intervjuer: #1 Var det et stort prosjekt det eller? #

Systemutvikler 1: #2 Det var svært ja, det er jo gammelt. 3år gammelt, så det var stort sett vedlikehold jeg drev med der. #

Intervjuer: #1 Men det er noe som Bedriften har utviklet tidligere? #

Systemutvikler 1: #2 Ja, der har gående supportavtale, så det kommer litt ting som skal endres underveis. # Seinere så var jeg og jobbet hos <Kunde>. De jobber med Shipping. Der var det noe vedlikehold og utvikling av nye applikasjoner.

Intervjuer: #1 var du da i Oslo eller? #

Systemutvikler 1: #2 Nei, da satt jeg her og jobba, var inne hos dem en gang i blant og fikk hilst på og litt sånn sosialt. men jeg har ikke jobbet for de der inne. #

Intervjuer: #1 Kjørte dere samme fremgangsmetode med rutiner på de andre prosjektene? #

Systemutvikler 1: #2 Nei, det var det ikke. På <Kunde>, det var jo mitt første møte med .NET. Og der var det, da hadde jeg prosjektlederen og arkitekten rett ved siden av meg

hele tiden. Så da var det veldig, prata mye med han da for å få hjelp og veiledning underveis. Og da var det liksom, prata veldig mye med han da underveis, og da var det liksom "så da må vi ha en funksjon der" så lagde jeg den. #

Intervjuer: #1 Du fikk direkte oppgaver fra her enkelt? #

Systemutvikler 1: #2 Ja # På det det prosjektet for <Kunde 2> så tok vi bare det fortløpende når det kom feilmeldinger. Så når det kom en feil, så jobbet vi med den.

Intervjuer: Kan du fortelle meg om arbeidsoppgavene du har her?

Systemutvikler 1: Nå er det jo utvikling, skrive kode.

Intervjuer: Har du noen spesielle rutiner du må gjøre når du skriver denne koden? Bruker dere noe system for å ta vare på koden, skrive kommentarer til den osv.?

Systemutvikler 1: Ja, vi har retningslinjer for hva som er ferdig kode, at det skal følge kodestandard, være kommentert og alt skal kompilere. Og det skal være enhetstestet. Så har vi en SourceSafe base som håndterer versjonskontroll og den type ting.

Intervjuer: #1 SourceSafe? #

Systemutvikler 1: #2 Ja, det er Microsoft sin Subversion. #

Intervjuer: #1 Er det noe Bedriften bruker generelt? #

Systemutvikler 1: #2 Nei, vi skal bort i fra det. Vi skal ha felles plattform for alt av versjonshåndtering. Jeg tror vi skal ha subversion på hele linja, for den skiller ikke på hva slags språk det er, den ser bare teksten. Så da får vi alt på en plass. #

Intervjuer: Hvordan velger du hvilke oppgaver du skal løse?

Systemutvikler 1: Det velger vi på iterasjonsplanleggingsmøtene. Så hver mandag planlegger vi hva vi skal gjøre denne uken. Og da setter vi opp hvilke brukerhistorier vi skal ha med i den iterasjonen, bryter de ned i konkrete oppgaver, og setter navn på dem, så jeg har ansvar for noen oppgaver og han andre utvikleren har ansvar for noe og designeren har ansvar for noe. De navnene vi setter på, er et ansvarsforhold, så hvem som faktisk gjør det, det blir jo ettersom hvilken arbeidsmengde en har akkurat der og da. Så hvis han gjør, har sett på en oppgave jeg har hatt ansvar for, så kan han fortsette med den, så kan jeg ta en lapp det gjerne står han sitt navn på. En plukker oppgaver når en er ferdig med noe annet, eller har dødtid. så hvis jeg sender noe over til han eller venter på noe dokumentasjon fra en eller annen plass, lisenser eller sånne type ting, så tar jeg så jeg ikke bare sitter og tvinner tommeltotter, så tar jeg en ny oppgave fra tavla, og flytter den over til "under arbeid".

Intervjuer: #1 Dere bruker et tavlesystem? #

Systemutvikler 1: #2 Vi bruker den Scrum-tavla som ble presentert på møtet. Så vi har gule lapper for brukerhistorier, og for hver brukerhistorie har vi X antall oppgaver. Så når vi tar og begynner på en oppgave, så tar vi den fra "ikke påbegynt" og setter den til "under arbeid" og når den er klar, setter vi den på "ferdig" #

Intervjuer: #1 Skjer det for alle oppgaver? #

Systemutvikler 1: #2 Ja #

Intervjuer: Om iterasjonsplanleggingsmøtet, der velger dere ut oppgaver, hvor velges disse oppgavene fra?

Systemutvikler 1: De er fra backloggen til hele prosjektet. Vi gikk gjennom brukerhistoriene helt i begynnelsen og hadde det som grunnlag for å estimere. Alle de brukerhistoriene har vi en lang liste på, så har kunden satt prioritet på dem, så hvilke vi lager tidlig og hvilke som er kritiske for at prosjektet skal bli bra og sånn. Så da har vi tatt

alt det som var veldig viktig og måtte ut tidlig, det tok vi ut først, så blir det mindre viktige oppaver etter hvert. Så alt det kritiske blir tatt unna først.

Intervjuer: #1 Skjer det at det blir omprioriteringer underveis? #

Systemutvikler 1: #2 Ja, det skjer det, og det står kunden helt fritt til å gjøre. Hele veien. Så at det som var høyt prioritert en uke, kan bli mindre prioritert neste gang. #

Intervjuer: #1 Så da er det sånn at hver uke plukker en de øverst prioriterte? #

Systemutvikler 1: #2 Ja for hver uke, er kunden med, og sier "OK, nå har jeg tenkt litt jeg vil ha med den og den istedenfor den vi snakket om tidligere" og vil ha de med i den iterasjonen her. #

Intervjuer: #1 Hva om, du ikke blir ferdig med en oppgave fra forrige uke? #

Systemutvikler 1: #2 Da følger den med i uka som kommer. #

Intervjuer: #1 Da blir den bare liggende? #

Systemutvikler 1: #2 ja. #

Systemutvikler 1: De brukerhistoriene som ikke blir gjennomført på den planlagte iterasjonen, de blir fulgt med til neste uke. Så ingenting blir flyttet ut av systemet før det godkjent, det er testet av oss, kunden har sett på det, testet det og godkjent det. Og da blir det satt til "verifisert" på tavla.

Intervjuer: #1 Skjer det noen ganger at dere går tilbake igjen på ting som er verifisert? #

Systemutvikler 1: #2 ja, nei, ikke verifisert. Der har vi ikke flyttet noe ennå. Om det kan skje det vet jeg ikke, men det er ofte det at vi har ting som står ferdig og er klar til test og alt skal være fint og flott, men blir flyttet tilbake til "under arbeid" for det kommer, kunden har endret sine kriterier eller funnet på at de vil ha tilleggsfunksjonalitet og sånne ting. # Så da blir brukerhistorien flyttet tilbake, og lagt på nye oppgaver.

Intervjuer: #1 Du snakket om at dere hadde visse krav til hvordan koden skulle være, hvordan har dere beskrevet det? #

Systemutvikler 1: #2 Det er en lapp som henger på tavla der inne, så den er veldig synlig hele tiden. Og der står det alle kravene til at vi skal kunne sette en lapp til ferdig. #

Intervjuer: #1 Hvordan fungerer kommunikasjonen innad i gruppa? Hvordan jobber dere sammen? #

Systemutvikler 1: #2 Veldig greit. Jeg for min del, synes det er veldig greit å sitte sånn som vi gjør. Altså hvis vi jobber med det, at vi sitter sammen. #

Intervjuer: #1 Dere sitter sammen? #

Systemutvikler 1: #2 ja det er vi tre, der inne som utvikler, vi sitter sammen hver dag og jobber. Da får en en god dialog hele tiden, mens en sitter og koder får en høre hva designeren tenker og da ser jeg hvordan en skal løse dette teknisk for å tilpasse designet osv. # Og så er det jo veldig greit å få utvekslet kunnskap, jeg har jo greie på noe og han andre vet noe bedre enn meg og så videre og så videre. Så da får en liksom ikke de der du stopper opp og har en lang vei til å få tak i hjelp.

Intervjuer: #1 Hvordan kommuniserer dere med prosjektleder? Hvis han ikke skulle være tilgjengelig. #

Systemutvikler 1: #2 Da ringer vi han, hvis det skulle være noe, som en ikke kan håndtere selv. #

Intervjuer: #1 Men han er stort sett tilgjengelig? #

Systemutvikler 1: #2 Han er veldig mye her oppe, ikke hele tiden. Sånn som nå er han fraværende, men som regel sitter han i naborommet. #

Intervjuer: #1 Men han deltar ikke i utvikling? #

Systemutvikler 1: #2 Nei #

Intervjuer: #1 Dere bruker daglige møter? Hver dag? #

Systemutvikler 1: #2 Ja, det fungerer veldig greit. En får en litt sånn ansvarsfølelse for hva en driver med. Både på godt og vondt, det er jo litt kjedelig å stå der og si "Nei den fikk jeg ikke ferdig fordi.. jeg ikke fikk det til". Ikke sant, men det er veldig greit også, da vet de andre, altså da får de beskjed om at "dette sliter jeg med". Trenger noe hjelp til å se på det. Så sånn sett er det veldig greit, og får et lite innblikk hva alle driver med. Og for prosjektleder og kunde og så er det nok veldig greit, da får en hver dag status på hva vi driver med og fremgang og den type ting. #

Intervjuer: #1 Det har skjedd flere ganger at du har fått hjelp etter et daglig møte? #

Systemutvikler 1: #2 Ja #

Intervjuer: Er det noen ganger prosjektleder har tatt tak i noe som har kommet opp som et problem på et daglig møte?

Systemutvikler 1: Nå må jeg tenke... Det vet jeg ikke, på de standupene er det stort sett programmeringsutfordringer og den type ting som vi har diskutert og da er det ikke så mye han kan bidra med. Bortsett fra å henvise oss til andre resurspersoner og det har jo skjedd. At han vet om folk i firmaet som kan sitte på svar til de problemstillingene vi har. Og da kommer jo det frem. "Dersom du lurte på det burde du ta en prat med han og han, for de vet det". Men ellers så er prosjektleder veldig flink til å ta tak i ting som ikke vi har kontroll på. Altså, han skal jo ha hoveddialogen med kunden. Så hvis vi trenger mye informasjon med dem, og ting som skal klareres og sånne ting, så tar han det. Og samme med drift av systemer sånn. Serveren og sånn, at alt det skal være oppe og kjøre, det ordnet jo han. Så han gjør jo det han kan for at vi kan fokusere på det vi faktisk skal gjøre der oppe.

Intervjuer: #1 Hender det at kunden kommer rett inn til dere noen ganger? #

Systemutvikler 1: #2 Ja det skjer det. #

Intervjuer: #1 Hva er det de spør om da? #

Systemutvikler 1: #2 De spør om endringer, ting som de ser som de ser som de vil ha litt annerledes. Eller ideer og den type ting. #

Intervjuer: #1 Og det pleier å løse seg greit eller? #

Systemutvikler 1: #2 Ja, vi tar det opp sammen med kunden, og setter opp nye oppgavelapper, hvis det trengs. For det er viktig for oss at vi har en god kommunikasjon med kunden. Slik at vi ikke er helt bundet til prosjektlederen for å få svar. #

Intervjuer: #1 Er det større oppgaver, eller er det av mindre art? #

Systemutvikler 1: #2 Det blir mer av det siste ja. "Den knappen vil vi ha der", og "det bildet der var ikke så fint". Det er mye design det går på der, for det er stort sett det de ser. #

Intervjuer: Selve prosjektet er det, stort sett websidegenerering, eller er det mye funksjonalitet som må programmeres også?

Systemutvikler 1: Det er endel funksjonalitet som må programmeres, det er det. Men mye av det går på å få det visuelle til å stemme. Med design og kundens krav, det er det jo. Men nå buker vi en ferdig CMS plattform i bunnen. Så vi skriver jo ikke dette helt fra begynnelsen. Men det er mye tilpasninger som må gjøres for å møte kundens krav.

Intervjuer: #1 Er det Bedriften sin egen CMS? #

Systemutvikler 1: #2 Vi bruker EPIServer. #

Intervjuer: #1 Det fungerer bra? #

Systemutvikler 1: #2 Jada, nå har ikke jeg noe grunnlag til å sammenligne det med så mange andre systemer, for det er det eneste jeg har jobbet med. Jeg synes det virker veldig greit. Jeg ser andre som har jobbet med andre CMSer er fornøyd og synes det er greit å jobbe med. #

Intervjuer: #1 Har det vært noen endringer på hvordan dere jobber underveis? #

Systemutvikler 1: #2 Nei #

Intervjuer: #1 Dere startet opp slik dere jobber nå? #

Systemutvikler 1: #2 Vi har egentlig det. De rutinene vi har.. Vi startet opp med daglige standup møter og iterasjonsplanlegging. Og det er slik vi holder på ennå. #

Intervjuer: #1 Dere har ikke endret antall daglige møter i uka eller noe lignende? #

Systemutvikler 1: #2 Nei # Det eneste som varierer litt er når vi har kjørt akseptansetester, når kunden er med. Det har variert litt på når kunden er har mulighet og sånn. Det var egentlig noe som... jeg vet ikke helt på at det skulle være fast eller ikke, men det har vært litt sånn når kunden har vært klar, og når vi har vært klar med koden. Når vi har tatt en del oppgaver som var klar til test og kunden har skrevet kriteriene så har vi tatt en bolk her og der.

Intervjuer: #1 Så det er ikke slik at akseptansetest kjøres hver uke? Hvor ofte kjører dere det? #

Systemutvikler 1: #2 Det har vel nesten... Første uke var det ikke noe, for da var det en del, det gikk jo litt tid i oppstart og.. alle skulle bli kjent med produktet vi jobbet på osv. Men andre uke hadde vi vel noe, og forrige uke kjørte vi. Og jeg lurte på om vi skal ha noe i morgen. # Men jeg tror det var snakk om at vi skulle ha faste tidspunkt på det og.

Intervjuer: #1 Hva er det som skal testes på den i morgen #

Systemutvikler 1: #2 Det vet jeg ikke. #

Intervjuer: #1 Er det noe som dere har jobbet med...? #

Systemutvikler 1: #2 Det er noe som vi har jobbet med forrige uke og denne uka. Så det blir de brukerhistoriene som står på "klar til test". #

Intervjuer: #1 Så det som ikke er testet, og det som er klart til test? #

Systemutvikler 1: #2 Ja, alt som er klart til test skal testes. #

Intervjuer: #1 Hvordan registrerer du hvor mye tid du bruker? #

Systemutvikler 1: #2 Det blir jo den tiden jeg sitter her oppe. #

Intervjuer: #1 Fyller du ut noe system på det? #

Systemutvikler 1: #2 jeg bruker, det er jo Bedriften som kjører prosjektet, så jeg bruker timeregistreringssystemet til Bedriften. Og da skriver jeg inn at jeg jobber på dette prosjektet, at jeg driver med utvikling, og at jeg har jobbet så og så mange timer, pr dag. #

Intervjuer: #1 For det er et eget timeregistreringssystem fra Bedriften? #

Systemutvikler 1: #2 Ja, og de fra det andre firmaet de registrerer timer i sitt system. Og da rapporterer til vår prosjektleder for hvor mange timer de har. For det blir jo fakturering fra to steder. #

Intervjuer: #1 Har dere noe system på hvor mye bidrag hver enkelt bidrag har til prosjektet? #

Systemutvikler 1: #2 Nei det gjør vi ikke. #

Intervjuer: #1 Bortsett fra hvor mange timer de produserer. Men ikke noe resultat av timene. #

Systemutvikler 1: #2 # I og med at han andre utvikleren sitter 30cm på siden av meg, får jeg vel ganske smekk om jeg ikke jobber. Og vice versa.

Intervjuer: #1 Hvem er han du jobber sammen med, er han mer erfaren enn deg? #

Systemutvikler 1: #2 ja, han var vel ferdig i 2003 og har jobbet siden da. #

Intervjuer: #1 Er det noe forskjellige typer oppgaver dere løser? #

Systemutvikler 1: #2 ja, han er mye flinkere enn meg på det som går på webgrensesnittet, altså det visuelle, få det på plass. Jeg er sterkere, eller jeg synes selv jeg er flink på det som går på de underliggende lagene, med datastrukturene og å få det til å henge på greip. #

Intervjuer: #1 Database og...? #

Systemutvikler 1: #2 ja der ligger jo mye klart, men den kommunikasjonen som må ligge mellom EPIserver og webgrensesnittet. Der gjør jeg mye. # Men nå en del, setter jeg opp veldig enkelt webgrensesnitt, så tar han mye av jobben med å få det stylumt og plukke sånn at det ser representabelt ut.

Intervjuer: Du kan starte med å fortelle litt om deg selv, hvordan stilling du har i dette prosjektet, hvordan stilling du har hos din arbeidsgiver og hvordan erfaring du har og sånt noe.

Systemutvikler 2: Ja, jeg har. I dette prosjektet jobber jeg som systemutvikler, og jeg har jobbet i <Samarbeidsbedrift> i tre år som systemutvikler. Først og fremst på Digimaker CMS. Og nå da på EPIserver, Bedriften. Jeg har en bachelorgrad i bunnen, med litt påbygging med litt mer teknisk utvikling.

Intervjuer: #1 <Samarbeidsbedrift>, hvor stort selskap er det? #

Systemutvikler 2: #2 Vi er 45 ansatte. Et reklamebyrå her i byen. Men det er jo et reklamebyrå da. I bunnen, så er vi vel 8-10 i min avdeling # Med utvikling og webdesign.

Intervjuer: #1 Du kom inn i dette prosjektet og fikk beskjed om at dere skulle bruke Scrum? #

Systemutvikler 2: #2 Ja, jeg har ikke jobbet med det (Scrum) før, jeg har hørt om det og lest litt om det og det er en fin måte å jobbe på # Vi jobber på en måte litt med samme prinsipper i <Samarbeidsbedrift>. For vi jobber mye med prosjektrom, og da er det tett dialog med kunden som er et av grunnprinsippene i Scrum Og små oppgaver, og ja. Men vi har ikke definert det som Scrum, mer bare vår egen måte

Intervjuer: #1 Det er vel slik folk pleier å gjøre det? #

Systemutvikler 2: #2 Ja, det hadde vært gøy og sett på om vi kunne skrive om måten å legge inn oppgaver på for eksempel, definere det som brukerhistorier og legge oppgaver på dem. #

Intervjuer: #1 For det er nytt for deg å bruke brukerhistorier? #

Systemutvikler 2: #2 Ja, det er jo helt greit. Tidligere har det hett "beskrivelse av oppgaven", men # Men det å få det inn i en riktig, eller en gjennomtenkt utviklingsmetode er jo fint. Og det føler jeg at Scrum er. Men jeg føler og at det har ganske mye overhead. Med møter og.. Iterasjonene. Kanskje hvis iterasjonene hadde vært lengre, at det hadde blitt bedre. Nå går jo en dag i uka med til møter omtrent. Og det, er litt voldsomt. Føler at en mister en dag med utvikling. Men, ja, vi må jo ha møter. Men kanskje ikke fullt så hyppig. Argumentet for å ha det hyppig var jo at vi ikke har jobbet i Scrum før. Så det er jo greit nok det. Men det er jo sånn at når vi kommer i gang med utviklinga, så skal vi ha et nytt iterasjonsmøte. Vi bør ikke ha så korte iterasjoner neste gang vi skal jobbe med Scrum.

Intervjuer: Sånn, med tanke på levering i slutten av iterasjonen og sånt noe, hvordan har det gått nå? Har dere levert det dere har planlagt?

Systemutvikler 2: Ehh, ja. De har jo glidd litt i hverandre, iterasjonene, men det har jo noe å gjøre med at de er korte. Også dette med at akseptansekriteriene har kommet seint, de har jo komme midt i iterasjonen, så sånn sett så har det glid litt over hverandre. Men vi holder jo flyten, så, det er bra.

Intervjuer: Når det kommer en ny oppgave, ut i fra endrede akseptansevilkår, blir det estimert med tanke på poeng med tanke på de nye oppgavene også?

Systemutvikler 2: Nei, det har ikke blitt det til nå, nå har det vært små oppgaver, og så har det vært mye resultat av akseptansekriteriene som har kommet seint. Så da er det jo per definisjon i de poengene som er estimert til brukerhistoriene.

Intervjuer: Føler du at du bruker, den mengde tid, poeng per oppgave samsvarer med hvor mye tid du bruker?

Systemutvikler 2: Ja, vi ligger jo på en 20 poeng i uka. Og det ligger vi stort sett gjevt på også. Noe riktig ligger det jo i poengene.

Intervjuer: Hva synes du om sammensetningen av deltagere i prosjektet?

Systemutvikler 2: Hovedsaklig, bra, vi kunne hatt en litt mer senior person på EPI, det hadde jo vært positivt.

Intervjuer: #1 Hvordan jobber dere i dag? Hvem tar hvilke ansvarsområder? #

Systemutvikler 2: #2 Det er jo <systemutvikler 1> og meg som har utvikleransvaret og <Designer> på design og front-end HTML til en viss grad. #

Intervjuer: #1 Det er kun dere tre som utvikler? #

Systemutvikler 2: #2 Ja, det har blitt til det. #

Intervjuer: #1 Er det planer om å ta inn andre, eller har det vært andre i prosjektet tidligere? #

Systemutvikler 2: #2 Nei det har ikke vært andre. Og jeg vet ikke om det kommer fler, det kommer an på noen samarbeidspartnere vi har. Men, bortsett fra det, så er det jo ikke noe sånn. Det er jo ikke noe kjempestort prosjekt heller.

Intervjuer: #1 Når hadde dere planer om å bli ferdige med dette prosjektet? #

Systemutvikler 2: #2 1. april er deadline, jeg tror vi klarer det. # Men det er jo på mange måter en levende side, med ny funksjonalitet hele tiden, det kan jo godt hende det blir videreutvikling av siden senere, men det vi har sagt til 1. april, det tror jeg vi klarer.

Intervjuer: Hvordan kommuniserer dere mellom, hvem som skal gjøre oppgaver, hvilke oppgaver som må gjøres, kan du forklare litt om det?

Systemutvikler 2: Det gjør vi jo på iterasjonsmøte. Da definerer vi opp oppgaver og hvem som skal gjøre de. På forrige møte så satt vi vel egentlig bare <Systemutvikler 1> og meg på alt. Så tar vi det bare mer eller mindre sammen.

Intervjuer: #1 Hvem bestemmer hva som skal gjøres? #

Systemutvikler 2: #2 Vi bestemmer det sammen. #

Intervjuer: #1 Dere tre utviklere? #

Systemutvikler 2: #2 Ja, eller kunden bestemmer jo hvilke oppgaver ho prioriterer så vi sammen med kunden, og Scrum masteren. Det er jo teamet som bestemmer. Men når iterasjonen er satt, så bestemmer vi jo det oss i mellom vi tre. #

Intervjuer: #1 Dere samarbeider stort sett om de fleste oppgavene, eller jobber dere en og en? #

Systemutvikler 2: #2 Det har nok vært litt det at vi ikke har samarbeidet så mye på oppgavene, at vi har gjort våre. Men nå prøver vi jo å samarbeide litt mer nå da. #

Intervjuer: #1 Det spørres kanskje på kompleksiteten i oppgaven også? #

Systemutvikler 2: #2 Det har vært ganske små oppgaver til nå. Sånn at det faller seg naturlig. Så har <Systemutvikler 1> vært litt syk og sånn i forrige uke, litt mye skole, så han forsvant jo mye i den iterasjonen #

Intervjuer: #1 Jeg forsto det sånn at det gikk greit, dere hadde klart dere greit allikevel. #

Systemutvikler 2: #2 Ja, han er jo junior, sånn at han må jo.. deretter liksom. # Ikke det at, han er jo flink på alle måter, men.

Intervjuer: #1 Er det sånn at dere sitter å leiter etter løsninger mye? #

Systemutvikler 2: #2 Nja, jeg har en kamerat på MSN som jeg sender litt meldinger. Så han hjelper meg litt når vi bare står fast med sånne små ting, men jeg kan ikke bruke han for mye. Han jobber jo i et annet firma i Oslo. Men det er greit å ha noen utenforstående å støtte seg til også. #

Intervjuer: Har dere fått noen kontakt i Bedriften som er gode på EPI også eller?

Systemutvikler 2: Ikke i Bedriften, men vi har jo en i EPI Norge som vi har mailkontakt med. Men mailkontakt går jo treigt, sånn at det er ikke så mye vi har fått brukt han. Vi har sendt han et par mailer. Men jeg synes at teamet, spesielt til å være første EPI prosjekt for alle i teamet, så synes jeg det fungerer bra.

Intervjuer: #1 Hva er det som skiller EPI fra Digimaker? Er ting vanskeligere? #

Systemutvikler 2: #2 For oss så er det jo vanskeligere, vi har jo et gigantisk komponentbibliotek til Digimaker som vi har utviklet i løpet av en tre års tid, sånn at det er mer klipping og liming i Digimaker prosjektene våre # Her er det mer sånn, ren utvikling fra bunnen av. Men det vil det jo være ved bytte av CMS til vi får bygd opp et komponentbibliotek på det også.

Intervjuer: #1 Det var vel en prosess som skulle settes i gang nå? #

Systemutvikler 2: #2 Jo det var det. #

Intervjuer: Sånn som med oppgaver dere løser. Hva er krava for at en oppgave skal være ferdig?

Systemutvikler 2: Nei det står jo i, eller en oppgave er jo ferdig når vi har gjort den på en måte. Men en brukerhistorie er ferdig når den oppfyller de kravene står på den hvite lappen på tavla. Så, det er jo egentlig sånn. Men, de små oppgavene, de flytter vi jo bort

Intervjuer: Vi hadde review møte på fredag, da kom det litt innspill til prosessen. Hva synes du om hva som ble sagt?

Systemutvikler 2: Jeg synes han virket veldig flink han <Navn> og det er litt gøy å treffe de folkene fra Bedriften også. De som har jobbet der lenge og jobbet med prosessen, den måten å jobbe på lenge har jo ofte litt fornuftige synspunkter. Jeg synes han sa mye bra. Han forklarte jo mye av det med de nye oppgavene som har vært en litt sånn ymse sak for min del. Som vi egentlig bare har skrevet på tavla og slengt de på. Men han var jo veldig klar på at en skal begynne på nytt for hver iterasjon. At det nødvendigvis følger fra iterasjon til iterasjon, altså det som ikke ble ferdig forrige iterasjon, at ikke det nødvendigvis skal dras over i neste. Og det er jo bra synes jeg.

Intervjuer: Kommer dere til å følge de rådene og aksjonene som ble satt der?

Systemutvikler 2: ja, det, var det reviewmøte han kalte det? Ja det må vi jo ta, det er jo fornuftig. Men jeg synes og vi burde, om ikke på dette prosjektet, så bør vi, vi trenger ikke et reviewmøte hver uke. For det skjer for lite på en uke til at det er nødvendig. Det er kjedelig å sitte og si "ja nå har vi laget den knappen der", men hvis vi hadde hatt et par uker så, så kunne kunden på en måte sett at det hadde skjedd litt mer. naturlig nok, vi er jo bare to utviklere, det er begrenset hva som skjer på en uke.

Intervjuer: #1 Det blir et lite miniprojekt dette, alt blir skalert ned. #

Systemutvikler 2: #2 Ja og så er det jo, for en 7-800 timer Scrum er kanskje litt overkill på et såpass lite prosjekt. #

Intervjuer: #1 Hva var estimatet, var det 500 timer? #

Systemutvikler 2: #2 Jeg har aldri sagt et estimat på 500 timer, kunden ønsket at det ble 500 timer, men # Men det var på en måte bare et ønske, det har aldri vært et tall jeg har vært i nærheten av å godkjenne. Og hvis en regner poeng pr. uke og ganger det med antall poeng i prosjektet, så kommer en til mye mer enn 500 timer. Så det tallet, står ikke jeg inne for, for å si det sånn.

Intervjuer: #1 Dere har tatt den tradisjonen med daglige møter, hva synes du om det? #

Systemutvikler 2: #2 Det synes jeg er greit, det er såpass kort også, at det har ikke noe å si. Det er greit å komme i gang hver dag kl. 9 # Det blir litt søkt, når standupen bare er meg og de to andre, sånn som det var i dag, da har vi bare gått gjennom hva vi hadde gjort, men når kunden og prosjektleder er der så er det veldig lurt.

Intervjuer: #1 Er ting dere sier der, er det enkle ting folk vet allerede, føler man at det blir gjentakelse? #

Systemutvikler 2: #2 Litt, men så blir vi jo obs på hva vi egentlig skal gjøre og hva vi har gjort. Vi må på en måte stå til rette for hva vi har gjort. Så jeg synes det fungerer fint. #

Intervjuer: #1 Har dere hatt noe sånt noe hos din arbeidsgiver? #

Systemutvikler 2: #2 Vi har jo statusmøter, men vi bruker jo ikke Scrum sånn at vi kaller det for sånne vanlige gammeldagse ting. Men statusmøter har vi i alle team som vi jobber #

Intervjuer: #1 Daglig? #

Systemutvikler 2: #2 Nei, som regel ikke daglig. Litt avhengig av hvor tett vi jobber, så har vi det en gang i uka. Statusmøte med oss og kunde, men det er jo litt gøy å sette sånne fancy titler på ting. # Men, altså, vi har jo aldri jobbet fossefall, som er motsetningen til Scrum, så jeg føler ikke det er helt fremmed å jobbe på denne måten, selv om vi ikke kaller det det samme. Vi har som regel jobbet etter en SRS. Altså en Kravspec, som har blitt skrevet på begynnelsen, men den er jo veldig sammenlignbar med at vi sitter i et møte på begynnelsen av Scrum og skriver brukerhistorier. I stedet for brukerhistorier, så skriver vi "Use case". En use-case er vel til og med en brukerhistorie. Men på, gamle use-case, ikke samme malen som vi gjør her. Vi bruker egentlig fossefallsmetodens use-case. Jeg vet ikke, vi kaller det use-case i kravspesifikasjonen også.

Intervjuer: #1 Men det er i utgangspunktet et brukersentrert måte å beskrive et problem? #

Systemutvikler 2: #2 Nei det er jo det som egentlig er forskjellen, vi skriver mer "Sånn er det". "Sånn og sånn skal det gjøres" og "Sånn og sånn blir det". #

Intervjuer: #1 Men det er vanskelig å skille metodene. Vi er også veldig åpne for nye oppgaver å ta inn med i hele kravspeccen #

Systemutvikler 2: #2 #

Intervjuer: #1 Men der kommer prosjektmøtet vårt inn igjen. Nye oppgaver kjøres inn i prosjektmøtet og så estimerer vi de der. Så vi håndterer de jo på en litt annen måte enn på tavla. #

Systemutvikler 2: #2 #

Intervjuer: #1 Hva trives du best i? #

Systemutvikler 2: #2 Jeg synes det er spennende å jobbe på denne måten, men effektiviteten er høyere på vår måte. Om det er fordi jeg er vant til å jobbe på vår måte, eller om det er fordi den er mer effektiv, må jeg gjøre noen fler scrumprosjekter for å vite. I forhold til at du diskuterer fram løsninger på papiret først, det tror jeg faktisk er en fordel. # Fordi at uansett hvordan en vrir og vender på det, så tar det tid med endringer i kode. Selv om man jobber i Scrum, sånn at en kombinasjon av å få spesifisert løsningen litt bedre på forhånd, så kanskje jobbe på team som vi gjør nå, kan bli bra, for eksempel at alle akseptanskriteriene til alle brukerhistoriene skrives. Altså, i forhold til vår kravspecc så blir det funksjonsbeskrivelsen skrives på begynnelsen, det tror jeg hadde vært en fordel. For da hadde man fra begynnelsen, både på designet og kodinga visst hvor det skulle ende.

Intervjuer: #1 For det som skjer nå, er at dere starter på iterasjonen, så kommer akseptanskriteriene i løpet av iterasjonen? #

Systemutvikler 2: #2 Ja, men nå har vi jo satt krav til at de skal komme senest tirsdag morgen. #

Intervjuer: #1 Hva skjer hvis dere starter på en oppgave, så kommer et akseptanskriterie som ikke tilsvarer det dere har begynt på? #

Systemutvikler 2: #2 Ja da må vi jo diskutere litt med kunden rett og slett. #

Intervjuer: #1 Kan det skje? #

Systemutvikler 2: #2 Ja, men vi er jo ikke helt fraværende fra hverandre, vi har jo en formening om hva det skal være. Men det er typisk litt sånn ekstra ting som kommer inn i akseptanskriteriene og sånt. # Sånn at, det blir litt nye oppgaver i løpet av uka. Det er akkurat de tingene, hvis en rett og slett hadde utvidet brukerhistoriene til å dekke funksjonsbeskrivelse også, så føler jeg vi hadde fått...

Intervjuer: #1 For da er det satt på forhånd kan du si? #

Systemutvikler 2: #2 Ja, men det er jo fossefall.. #

Intervjuer: Scrum-master i prosjektet, hva er han sine oppgaver?

Systemutvikler 2: Han har jo en prosjektleders oppgaver. Det innebærer å holde flyten i prosjektet, skrive statusrapporter og.. prosjektlederting (latter) Jeg er ikke prosjektleder. Han holder oss og kunden i tøylene. Men det synes jeg fungerer, han er flink. Han ordner opp i problemer.

Intervjuer: Har dere noen utfordringer med tanke på kommunikasjon?

Systemutvikler 2: Jeg føler vi gjør det vi mener, og hører det vi trenger. Nei, jeg føler ikke vi har noen kommunikasjonsproblemer i vårt team.

Intervjuer: Kan du forklare hvordan en akseptansetest blir utført?

Systemutvikler 2: Da tar vi brukerhistoriene, og akseptanskriteriene, så viser vi egentlig brukerhistoriene på skjermen. "Fort og gæli" Det har vel også med at vi må korte ned møtetiden og hvis hun vil kikke mer på det så kan hun gå inn og kikke på det. Men det har hun aldri tid til.

Intervjuer: #1 Hva skjer hvis kundene ikke er enig med at et akseptanskriteriene ikke er nådd? Skjer det? #

Systemutvikler 2: #2 Det har skjedd flere ganger, det er jo litt av det vi pratet om på reviewmøtet, at han (oppgaver) skal egentlig ikke gå automatisk videre, for vi har som regel akseptansetestene sent i uka. # Sånn at vi bare må få notert de ned, på en sånn blå lapp vi har tatt i bruk nå. Så vi ikke mister de. Så når vi diskuterer på mandagen om vi skal ta de videre i den nye iterasjonen. Men det har vi jo ikke gjort til nå. Nå har vi jo bare hengt på den lappen, så har den bare automatisk blitt med videre.

Intervjuer: #1 Så nå kommer dere til å gjøre det på den nye måten? Med en blå lapp? #

Systemutvikler 2: #2 En blå lapp er fint (latter) Det er jo et poeng uansett, selv om jeg kanskje ikke bryr meg så hard om det glir i hverandre, så er det jo et poeng å tømme tavla av og til. # Det ser jo veldig fint ut

Første intervju med Scrum-master prosjekt 3

Scrum-master: ja, jeg har vel 10 prosjekter kanskje eller noe sånt noe

Intervjuer: Skal vi se; det jeg tenkte vi kunne gjøre i dag, jeg må komme i gang med hele datainnsamlingsprosessen og sånt noe, så da tenkte jeg at vi kunne ta et første intervju nå så, vi bare får kommet i gang og jeg får litt bakgrunnsstoff rett og slett

Scrum-master: Ja

Intervjuer: Så må jeg nok komme igjen og spørre om jeg kan ta en samtale til med deg i løpet av..

Scrum-master: #1 Ja det skal vi nok ordne. #

Intervjuer: #2 Det blir mye nå den måneden her, og de neste fire ukene blir det stort sett datainnsamling på meg. # Så da håper jeg på at jeg kan få være så mye som mulig her nede også.

Scrum-master: Vi har tenkt fire prosjekter du egentlig kan følge nå. Da er det <Prosjekt 3 Kundenavn>-prosjektet; det foregår i Oslo.

Intervjuer: #1 Ja men det skal vi.. #

Scrum-master: #2 Så er det <Navn1> sitt prosjekt på <Prosjekt 1 Kundenavn> # Så er det <Navn2>. Han er ferdig med å sette opp smidig utvikling ute hos <Prosjekt 2 Kundenavn>. Vi hjelper med å innføre Smidig utvikling der. Så han kan du jo følge litt. Så er det jo <Kundenavn 2>.

Intervjuer: Hva gjør du på <Kundenavn 2>?

Scrum-master: Der er det en virksomhetsportal. #1 Så er det, ja det er vel det. #

Intervjuer: #2 Vi kan bare starte, kan du bare fortelle deg litt om deg selv, din stilling og sånn så jeg har det på tape. #

Scrum-master: Jeg heter <Scrum-master Navn>og jobbet som prosjektleder i <Bedriften> og har tatt på meg det faglige ansvar for Smidig metodikk.

Intervjuer: Du er prosjektleder for et av 10 prosjekter som pågår nå?

Scrum-master: #1 Jeg er prosjektleder for Customer Business Solutions hos <Prosjekt 3> Kundenavn>. #

Intervjuer: #2 Kan du fortelle litt mer om det prosjektet? #

Scrum-master: Det er et prosjekt for innføring av et havneadministrasjonsverktøy, der kunden skal kunne gå inn og registrere det vi kaller "Port Calls". De skal sette opp hvilke aktiviteter som skal utføres på skipa, og sånt noe #1 Hva som skal lastes og losses i de ulike havnene, tracking av kostnader tilknyttet det, så skal vi ha en Cash management løsning inni her. #

Intervjuer: #2 Javel? # Hvor mange personer er det som er på det prosjektet?

Scrum-master: Fra <Bedriften> så er vi 6 nå, 2, nei 3 stykker fra <Prosjekt 3 Kundenavn>, så er det en fra <Samarbeidsbedrift>

Intervjuer: #1 Er det utviklere som er også fra.. #

Scrum-master: #2 Nei, det er fire utviklere, så er det tre på en måte kravstillere, så er det en han som er fra <Samarbeidsbedrift>, han er kundens prosjektleder. # #1 Så er jeg prosjektleder for gjennomføringen fra <Bedriften> sin side. #

Intervjuer: #2 (Innledning til min master) # Så jeg har forstått det som om dere nå bruker Scrum, eller har dere tilpasset Scrum på en spesiell måte?

Scrum-master: Nei, altså vi bruker hovedsaklig Scrum, men det vil alltid være en sånn tilpasning i en eller annen form for å finne en måte vi skal jobbe på for å få det til å funke best mulig for oss. Så sånn vi liker egentlig ikke å kalle det Scrum, men det er jo egentlig

Scrum. Det vi har lyst til å bygge opp er en verktøykasse, som vi kan tilpasse hvert enkelt prosjekt, som prosjektlederen kan gå og plukke elementer fra, som passer inn i akkurat det prosjektet.

Intervjuer: #1 Er det noe som blir gjort i alle prosjekter, eller er det bare noen prosjekter som bruker den type verktøykasse som dere har nå? #

Scrum-master: #2 Altså det er en vurdering for hvert enkelt prosjekt. I den størrelsen og de prosjektene som vi tar på oss nå er nå, er Scrum en fin måte å gjennomføre prosjektene på, men det er klart blir det mye mye større. Og der vi må bryte ned i flere Scrum team, kan det være aktuelt med andre smidige metodikker. # #1 For eksempel Agile Rup. #

Intervjuer: #2 Hvem er det som har tatt den beslutningen om at <Bedriften> skal bruke smidig metodikk? #

Scrum-master: Det er jo en beslutning på ledelsesnivå i <Bedriften>.

Intervjuer: #1 Så det er retningslinjer der følger? #

Scrum-master: #2 Ja. Bakgrunnen for den beslutningen er at vi får et mye tettere samarbeid med kunden, at kunden er mye mer delaktig i prosjektene og er stadig med og prioriterer og forklarer og alt sånt noe. Pluss at vi etter at prosjektet er gjennomført, så står vi med et bedre tillitsforhold. #

Intervjuer: Du kan jo fortelle litt om <Prosjekt 3 Kundenavn> prosjektet ditt nå, kan du beskrive prosessen frem til nå?

Scrum-master: Nå startet prosjektet på onsdag, så de to siste ukene har vi bare speccet opp prosjektet. Så det vi har gjort da er å bygge opp den prosjektbackloggen, vi har gått gjennom andre applikasjoner som de ønsker seg funksjonalitet fra inn i det systemet her og formulert det som brukerhistorier. Vi har bestemt at det skal være 14 dagers iterasjoner. Så vi har designet brukerhistoriene ut i fra den lengden. Vi har, det er et ganske stort prosjekt, så vi har brukt så og si en uke på kartlegge alle brukerhistoriene. Og da har hele prosjektteamet vært til stede, eller ikke hele, en utvikler, en arkitekt, tre kunderepresentanter og meg. Dokumentert det med både avgrensninger og forutsetninger sånt noe, og så har vi brukt to dager på å estimere det, med "planning poker".

Intervjuer: #1 Så dere har hele lista klar for alt som skal være med? #

Scrum-master: #2 Ja, det har vi. Og vi har også gjort en initial prioritering av alle brukerhistoriene. # Så da skal vi kjøre iterasjonsplanlegging i morgen. Og da er prioriteringa så og si klar. Og da er det bare å viderefødre den backloggen, bryte den ned i elemen.. brukerhistorier vi tar inn i hver iterasjon og bryte dem ned i oppgaver og estimere i timer.

Intervjuer: #1 Det blir en og en brukerhistorie per iterasjon? #

Scrum-master: #2 Det blir flere, vi skal etter hvert estimere "fart", hvor stor fart vi klarer å ha. Men det blir flere brukerhistorier #

Intervjuer: #1 Du nevnte Planning Poker, hva er det for noe? #

Scrum-master: #2 Det er en metode for å estimere opp, altså vi tar utgangspunkt i en # brukerhistorie fra backloggen, som vi tror ligger på snittet i kompleksitet og omfang. Så den setter vi til en femmer. Vi har en skala som er en, to, tre, fem, åtte, tretten og tjuen. Og plukker ut den vi tror er en femmer og så vekter vi alle brukerhistoriene ut i fra den. Så har funnet ut at det er en veldig effektiv måte å estimere opp. En menneskehjerne er skapt, fungerer bedre på å vekte ting opp i forhold til hverandre enn å si at "dette tar ti timer" for eksempel.

Intervjuer: #1 Hvor har dere fått det fra? #

Scrum-master: #2 Det er en del av Scrum. Så bryter vi da ned hver enkelt brukerhistorie, og kan vi klare å si hvor mange timer ett poeng er. #

Intervjuer: Hva med.. #1 Nå har jo ikke dere startet utviklingen ennå, har dere fulgt opp det med daglige møter nå også? #

Scrum-master: #2 Nei, Det først når teamet blir satt på. Vi kommer til å kjøre daglige Scrum-møter. Utviklerne er i Oslo. Vi har fått ett møtelokale som vi gjør om til et prosjektkontor, så der sitter alle rundt et bord og vi har tavler som prosjektoppfølgingsverktøy. #

Intervjuer: #1 Hva slags, hvordan kontrollerer dere kvaliteten på hva som blir laget? #

Scrum-master: #2 Kunden kommer til å designe testkriterier, og de kommer til å bli gjennomført før slutten av hver iterasjon. # Så vi har en kvalitetssikring da innenfor hver iterasjon, men også mot slutten, når alle iterasjonene er ferdig, så kjører vi et testløp for å se at alt spiller ordentlig sammen.

Intervjuer: Har dere lagt opp til noe dokumentasjon for brukerne? Eller er det noe som kommer senere?

Scrum-master: For at en utvikler skal kunne si at en brukerhistorie er ferdig, så er et av kriteriene at koden skal være dokumentert og at det han har laget skal være dokumentert. Hvis ikke får han ikke godskrevet poengene sine i den iterasjonen. #1 Men vi har ikke kommet så langt at vi har sagt at koden skal se "sånn og sånn ut" og sånt noe, men det kommer. #

Intervjuer: #2 Men dere har speccet opp visse retningslinjer? #

Scrum-master: Ja, det har vi i alle prosjektene vi har. For at de skal kunne sette en brukerhistorie eller en oppgave til avsluttet, så skal visse kriterier være gjort. Så det er sånn. Koden skal være dokumentert, han skal være sjekket inn, det skal være utført akseptansetest osv.

Intervjuer: #1 Det prosjektet her, er det sånn det blir gjennomført i alle prosjekter fremover torr du, i samme størrelsesordenen? #

Scrum-master: #2 Ja, det tror jeg. Så lenge vi kommer til å fortsette å ha gode erfaringer med det, så kommer vi til å gjøre det. #

Intervjuer: Er det noen prosjekter som ikke følger denne fremgangsmetoden?

Scrum-master: Nei ikke som jeg kan komme på faktisk, de fleste bruker en eller annen form for iterativ utvikling. <Kundenavn 3> prosjektet er jeg usikker på i hvor stor grad gjennomfører det her. #1 Det er ingen som kjører fossefall i hvert fall. #

Intervjuer: #2 Jeg leste gjennom. Det er noen tidligere masteroppgaver som har jeg tror de har pratet med dere om agil utvikling, har du vært med på noe intervju tidligere? # Har du hentet inspirasjon og tatt inn noen prinsipper fra andre metodikker enn fra Scrum?

Scrum-master: Nei, altså, det går mer på erfaring. Det er ikke noe sånn at jeg har plukka, jeg har jobbet en del med RUP, men jeg kan ikke si at jeg har tatt med meg noe derfra, men det ligger på en måte i hvordan du tenker og sånn da. Altså den metoden vi har bygd opp nå er så og si utelukkende Scrum.

Intervjuer: Er det noe annen dokumentasjon som kan være nyttig for meg å se på?

Scrum-master: Det skal vi nok fikse, vi har et sånt prosjektstyringsverktøy, hvor en av elementene er backloggen og vi deler da opp i sprinter, så det blir da et regneark, et stort regneark. Vi tar og deler opp i sprinter, og sprinten inneholder da oppgaver. Og ut i fra det så bygger vi opp en "Burn Down" og vi ser hvordan farta utvikler seg i løpet av de ulike iterasjonene og så videre.

Intervjuer: #1 Det er et levende dokument som hele tiden eksisterer? Det gjelder også på <Scrum-master, prosjekt 1>s prosjekt (<Prosjekt 2 Kundenavn>)? #

Scrum-master: #2 Ja, det er det. Det verktøyet blir i alle fall brukt på de to prosjektene. Tror ikke det blir brukt på <Kundenavn 2> og <Prosjekt 3 Kundenavn>. # Det kan nok bli brukt <Prosjekt 3 Kundenavn>, men jeg tror <Kundenavn 2> kjører sitt eget løp. Det prosjektet har pågått over lengre tid, det er fase 3 som pågår nå De har hatt god erfaring med de to andre fasene, så jeg har ikke innført noe nytt. <Navn3> sitter som ansvarlig på det prosjektet.

Intervjuer: #1 Kommer du til å være i Oslo mye i tiden fremover med tanke på at prosjektet er der? #

Scrum-master: #2 Ja, det blir ukentlig # Jeg kommer til å basere meg på en dag i uka foreløpig, og ser hvordan det går, hvis ikke blir det to.

Intervjuer: #1 Hvordan gjør dere det på daglige møter? Kommer dere til å ta det på telefon? #

Scrum-master: #2 Ja, gruppa kommer til å sitte sammen der inne, så kommer jeg til å bli sittende her nede på telefon. #

Intervjuer: #1 Hvordan fungerer det? #

Scrum-master: #2 Vi har prøvd det på et annet prosjekt på <Prosjekt 3 Kundenavn>? Det fungerer det. #

Intervjuer: #1 Klarer man å holde seg på 15min? #

Scrum-master: #2 ja der er vi beinharde, Alt utenomstakk og slikt blir tatt etterpå. # Vi prøver å være flinke på å avtale. Altså havner vi i en diskusjon, så avtaler de to eller tre et møte senere.

Intervjuer: #1 Hvordan er det som prosjektleder når en bruker Scrum, er det ikke slik at prosjektlederen skal så ansvarlig for å fjerne hindre for prosjektteamet? #

Scrum-master: #2 Nå er det ikke noe som heter prosjektleder, det er jo Scrum Master. En av oppgavene er å fjerne hindre ja #

Intervjuer: #1 Er det merkbart at det krever med av deg som Scrum master enn den generell prosjektlederstilling ellers. #

Scrum-master: #2 Nei, en prosjektleder i et fossefallsprosjekt også har oppgave i å gjøre så utviklingsteamet kan fungere optimalt. Jeg sanser ikke det som noe forskjell.

Forskjellen er kanskje i at du har mer fokus på det i et Scrum prosjekt enn i et Fossefallsprosjekt. Men det er samme filosofien som ligger bak. #

Intervjuer: #1 Bruker dere Scrum master i stedet for som prosjektleder som navn? #

Scrum-master: #2 Ja, både og. # Prosjektlederrollen er mer kjent. Hos <Prosjekt 3 Kundenavn> bruker vi prosjektleder, men vi introduserer Scrum Master, så de er klar over at det heter det.

Intervjuer: #1 Så kunden begynner å bli vant til hvordan dere jobber på? #

Scrum-master: #2 Det her blir jo det første store prosjektet vi kjører hos <Prosjekt 3 Kundenavn> som går på Scrum. Vi har kjørt noen små prosjekter tidligere som har på en måte vært innenfor en avdeling. Men det her er såpass stort at det kommer et hakk høyere opp i organisasjonen da, og der er de vant til å kjøre fossefall. # Og det blir en stor utfordring. Det er mange som vil kjøre fossefall og er veldig skeptiske til smidig utvikling.

Intervjuer: Er det berettiget å være skeptisk?

Scrum-master: Nei jeg synes ikke det, ting er mer transparent i prosjektet. Vår erfaring er når de først har blitt kjent med måten å jobbe på, foretrekker samtlige det. Men, det de er engstelige for er på en måte det, du utsetter den detaljspeccinga til hver iterasjon og du får ikke en fast pris "up front". #1 de tilbuda vi får inn nå, alle vil ha en fastpris da. #

Intervjuer: #2 Det er kanskje vanskelig i en anbuds konkurranse og? #

Scrum-master: Det er det vi forsøker å forklare, at selv om du gir en fastpris "up front" så er det i fåtall av tilfellene at du klarer å treffe Du klarer ikke å, kunden klarer ikke beskrive hva han vil ha, og samtidig klarer ikke vi å tolke det han beskriver på riktig måte. Altså det er mye bedre at vi samarbeider om løsningen. At vi tar det trinnvis i form av iterasjoner.

Intervjuer: #1 Hvorfor har dere valgt å ha iterasjonstiden til to uker på <Prosjekt 3 Kundenavn>?

Scrum-master: #2 Vi ønsker å vise leveranser hyppigere, og få tilbakemeldinger raskere. Generelt fire uker er alt for langt.

Intervjuer: #1 Er det noen erfaringer som dere har gjort? #

Scrum-master: #2 Ja, vi ønsker å få tilbakemeldinger på det vi gjør, hele tiden, så vi kan justere oss. # Vi synes det er relativt liten overhead ved å gjøre det.

Intervjuer: #1 Får utviklerne jobbe i fred, er det mye endringer i løpet av en sprint? #

Scrum-master: #2 Nei, innenfor sprinten er det ikke... Det kommer an på lengden på de selvfølgelig. Men vi prøver å være relativt strikte på innholdet av en sprint, når den er satt, skal vi gjennomføre det. Kommer det nye ting, tar vi det inn i backloggen og prioriterer det som en endring. # Det med at utviklerne blir forstyrret, det er en stor utfordring egentlig. Ofte kommer kunden og stiller spørsmål, på om andre ting også da. Men jeg tror vi har klart å isolert <Prosjekt 3 Kundenavn> teamet. Vi har satt dem i et annet bygg i forhold til IT-resursene. Det er et bevisst valg for å isolere de lengst fra de. Vi har jo flere systemer vi har utviklet for de tidligere, så mange kommer inn til de og spør spørsmål om andre ting en selve prosjektet. Så vi har distansert de i et annet bygg for å isolere de mer. Men det blir en utfordring å kanalisere sånne ting gjennom både kundes prosjektleder og meg da.

Intervjuer: #1 Hvem er det som har ansvaret for backloggen, er det dere i felleskap? #

Scrum-master: #2 Ja, jeg kommer til å ha ansvaret for å oppdatere den og sånt noe, men det blir jo tatt opp hvis det er noe så blir det tatt opp på hvert scrum-møte. #

Andre intervju med scrum-master i prosjekt 2

Intervjuer: #1 Hvordan har det gått de siste to ukene? #

Scrum-master: #2 Det har gått veldig greit. Ferdig planlagt, jeg husker ikke hvordan status var sist når vi pratet sammen, men vi er midt i første iterasjon nå # Og det går greit, alltid oppstartsproblemer og sånt noe med først og fremst miljø osv. Bortsett fra det, er prosjektet satt opp riktig og det fungerer sånn foreløpig i alle fall.

Intervjuer: #1 Hvor lange iterasjoner hadde dere nå? #

Scrum-master: #2 To uker, vi startet opp på onsdag, så da er første uke i første iterasjon ferdig i går. #

Intervjuer: #1 Så dere ligger litt etter, siden det er noe problemer med miljø? #

Scrum-master: #2 Nei jeg tror det går greit. #

Intervjuer: #1 Og alt har gått etter planen slik du beskrev det forrige gang? #

Scrum-master: #2 Ja vi kjørte daglige møter hver dag, har faste møter halv 12. #

Intervjuer: #1 Er det noen spesielle utfordringer som har kommet opp? #

Scrum-master: #2 Det er knyttet til miljø, at utviklingsserveren står her i Kristiansand. De jobber med VPN og sourcesafe og det har ikke fungert helt optimalt. De jobber med doble VPNtuneller og sånt noe. # Har hatt problemer der da, men der skal vi over på TeamFoundation ser det ut til.

Intervjuer: #1 Dere bruker sourcesafe, det samme som de bruker på <prosjekt 1> nå? #

Scrum-master: #2 Ja #

Intervjuer: #1 Det er det Microsoft systemet? #

Scrum-master: #2 TeamFoundation er det nye i alle fall fra Microsoft, vi skal over på det. #

Intervjuer: #1 Dere skal over på det i løpet av det prosjektet her? #

Scrum-master: #2 ja egentlig så fort som mulig, nå er sourcesafe et stort "issio", vi bruker tid på det daglig for å fikse ting. #

Intervjuer: #1 TeamFoundation er noe hele Bedriften skal over på? #

Scrum-master: #2 Det er en beslutning om at vi skal bruke et system, men den beslutningen er ikke tatt ennå. Siden vi har blitt Microsoft Gold partner, så ligger det litt i kortene at vi skal bruke det. #

Intervjuer: #1 Og det ser bra ut? #

Scrum-master: #2 Ja de gjør det, det eneste issioet vi har med det er at vi er usikre på hvordan det fungerer med Java. Vi kjører tester og sånn der nå. #

Intervjuer: #1 Microsoft er kanskje ikke så glad i Java? #

Scrum-master: #2 nei hehe #

Intervjuer: #1 I <Kundenavn> var det slik at kunden ikke jobber i Scrum? #

Scrum-master: #2 # <Kundenavn IT avdeling> , de jobber i fossefall, men vi er på en måte tatt inn som en del av <Kundenavn organisatorisk enhet>. Så vi har ikke noe direkte tilknytning til IT, bortsett fra at vi har masse motstandere. For vi har noe med de å gjøre med det allikevel, for vi har leveranser inni et program, Som avhenger av blant annet < Kundenavn IT avdeling > at de skal gjøre ting.

Intervjuer: #1 Hvordan motstand er det dere får? #

Scrum-master: #2 Nei altså, de har prøv Scrum tidligere, men de synes ikke noe om det. #

Intervjuer: #1 Men det er ikke sånn at de motarbeider dere? #

Scrum-master: #2 Jo, det er politikk på høyt nivå her altså, her legger de kjepper i hjula konstant. #

Intervjuer: #1 har du noen eksempel på det? #

Scrum-master: #2 Nei, de har problemer med, altså vi er en del at et program ikke sant. Også et annet prosjekt, der er vi avhengige av et prosjekt. Og de styrer, vi har et issio med et system som vi skal bygge videre på da. Som er basert på Oracle, men vi skal bygge på Microsoft, eller SQL server. Og det er en beslutning de må ta om de skal konvertere, om vi skal bygge på Oracle også på det nye, eller om vi skal ja.. Og det er en sånn treneringsprosess der da. # Og det er en deadline nå før påske, hvis vi ikke lykkes å nå den da, så er vi blåst for å si det sånn.

Intervjuer: #1 Har de fått den deadlineen også? #

Scrum-master: #2 Ja det er en deadline som hele programmet må forholde seg til da. Og de ligger an til å ikke klare den da, men på en måte de er , ettersom de er internt så er de litte granne beskyttet da. #

Intervjuer: #1 Det er ikke så lett med andre ord.. #

Scrum-master: #2 Nei, det er ikke det. Kundens prosjektleder, han driver kun med politikk. Jeg har alt det administrative prosjektorganisering og sånt noe, så har han møter og ordner alt det politiske da. # Det hadde ikke fungert at jeg sitter her i Kristiansand da...

Intervjuer: #1 Det er noe som kan få hele prosjektet til å stoppe opp, og det er kun fordi dere bruker Scrum? #

Scrum-master: #2 nei... #

Intervjuer: #1 Liker de ikke Bedriften i det hele tatt eller er det? #

Scrum-master: #2 Nei altså jeg vet ikke... jeg er for ung i <Kundenavn> til å forstå hvorfor, men metodikk er nok en av tingene. #

Intervjuer: #1 Unge friske Bedriften kommer og skal fortelle hvordan en skal gjøre ting? #

Scrum-master: #2 Ja, altså de som jobber på it og de som styrer, har vært i militæret i x antall år og er i 50-60åra, og de er ikke så lette å snu. #

Intervjuer: #1 Hvordan spesifiserer dere hvor mye timer dere bruker i prosjektet? #

Scrum-master: #2 # Vi, hver uke må alle resurser legge ut hvor mange timer de bruker pr brukerhistorie. #

Intervjuer: #1 Hvor registreres det? #

Scrum-master: #2 Det er et Excel opplegg som de registrerer for hver uke. Og så tar jeg det inn i prosjektstyringsverktøyet. Så utarbeider jeg da fire forskjellige skårekort ut i fra det. #

Intervjuer: #1 Du bruker Balanced Scorecard? #

Scrum-master: #2 Ja det er på en måte det. Excel dokumentet er det samme dokumentet som hadde burn-down og brukerhistoriene. # De fire scorekortene vi bruker, det ene går på scope, hvor vi ser hvor mange originale scope. Om det er kommet nye godkjente endringer i løpet av iterasjon. Om det er kommet noen avslåtte endringer i løpet av iterasjonen. Og så faktisk opp (utydelig) den ene. Så går det på burn-down. og forventet burn-down, planlagt burn-down og sånn. velocity og resurser. Og i resursene kommer da inn forventet bruk, hvor mange timer vi har brukt, varians så vi kan måle overhead og sånt i prosjektet.

Intervjuer: #1 Er det noen av de andre prosjektene som gjør dette? #

Scrum-master: #2 Nei #

Intervjuer: #1 Hvorfor bruker dere det? #

Scrum-master: #2 Fordi jeg har laget det... Men det er noe jeg har laget for dette prosjektet her #

Intervjuer: #1 Hvordan ser det ut nå? Fungerer systemet ditt? #

Scrum-master: #2 Nei vi er jo ikke ferdig med første iterasjon engang, så det er litt tidlig å si noe om det. #

Intervjuer: #1 Utviklerne har klart å følge opp hvor mye tid de har brukt. #

Scrum-master: #2 ja de fyller bare ut et ukeskjema, for uke x har de x antall timer pr dag og så pr brukerhistorie, så har vi noen standardkoder, administrasjon, planlegging, nedetid på systemer og sånne ting. #

Intervjuer: #1 Bruker dere Bedriften sitt timeregistreringssystem i tillegg eller? #

Scrum-master: #2 Ja, det gjør vi. # Den styrer faktureringa. Grunnen til at vi gjør det sånn er at Visma ikke er fleksibelt nok til å håndtere et prosjekt.

Intervjuer: #1 Du har funnet et behov for et prosjektstyringsverktøy? #

Scrum-master: #2 Visma er jo ikke et prosjektstyringsverktøy, men jeg skulle veldig gjerne hatt et web-basert system for det eller noe sånt noe. # Men altså, Excel er bra det altså, du har voldsom frihet til å ha kontroll på det. Du blir fort veldig låst på et web-basert system. En må nødt til å ta ut de og de rapportene og sånt noe.

Intervjuer: Kan du forklare litt om din stilling i Bedriften

Scrum-master: Min historikk, jeg begynte jo i august i fjor (2007) og så har jeg vært tilbakeleid et halvt år til min gamle arbeidsgiver. Jeg har jo sånt sett i praksis kommet inn i rullene i januar. Jeg er vel egentlig ansatt som, jeg er ikke helt sikker på hva stillingstittelen om det er seniorkonsulent eller hva det er, i praksis har jeg endt opp med å jobbe med litt av hvert disse to månedene jeg har vært ute i felten for å si det sånn. Jeg har jobbet en del med tilbud, jeg har, nå sitter jeg vel på tre, fire roller kanskje akkurat nå. En hvor jeg har, er ute hos kunde nå, og hjelper dem med å få opp Scrum som arbeidsprosess ute i bedriften Der er jeg rett og slett bare en innleid Scrum-master for å si det sånn.

Scrum-master: Det er et litt spesielt prosjekt, det er ikke sikkert at de i denne runden får oppfølging i mer enn en iterasjon. De valgte et prosjekt, hvor de fikk prosjekteier med på dette og se om det er noe som kunne passe for oss (dem). # Så vi har kjørt det helt vanlig helt, helt vanlig fra starten av. Selv om vi kanskje blir sammen mer etter en iterasjon. Så vi har kjørt brukerhistorier og det løpet der som jeg bruker. Og så har jeg et annet prosjekt nå, som jeg skrev tilbudet, <Prosjekt 4>, og det har jeg og fått nå. Jeg må ha med meg noen fler på det, men <overordnede> har sagt at jeg skal ta det. Så da ble jeg Scrum-master på det også.

Intervjuer: #1 Hvor stort prosjekt er det? #

Scrum-master: #2 Det er et utviklingsprosjekt hvor vi skal utvikle et miljørapporteringssystem for <Prosjekt 4>. De har hatt det ute på anbud, og det har vi vunnet. Det er et utviklingsprosjekt. Hvor stort omfang det er, 450timer eller noe sånt noe. Det skal leveres 1. mai. # (Mye støy uklart - pause)

Intervjuer: Hva har jeg nevnt nå? Jeg har nevnt <Prosjekt 2>, jeg har nevnt <Prosjekt 4>. Jo og så fungerer jeg litt som QA på <Prosjekt 1> Og det var.. Hovedsaklig så skal jeg hjelpe de med å dra ut litt samkjøringseffekter mellom flere prosjekter de har. Altså prøve å komme i gang med å trekke ut komponenter. I tillegg så skal jeg hjelpe de litt bare at vi har definert opp at vi skal hjelpe de litt med prosessen for å få den gåands mest strømlinjeformet som mulig. Jeg hadde jo første møte med de på fredag som jeg var med på og da begynte jeg å nøste litt. #1 Når vi snakker om det prosjektet (<Prosjekt 1>) Er det noen spesielle utfordringer du ser der? #

Scrum-master: #2 Jeg har ikke vært så mye inne i det ennå. Jeg har vært QA på det en stund, men jeg har liksom aldri følt jeg har fått mandatet så når jeg og <Scrum-master, prosjekt 1> fikk satt oss ned her for en ukes tid her så sa vi; ok, jeg skal ha en QA rolle, hva er det jeg egentlig skal gjøre i praksis. Så satt vi opp noen punkter. # Så har jeg fått en konto, som jeg fører på, og da er det greit. Da har jeg et mandat til å gjøre noe, så da begynte jeg jo med det. Og jeg har jo hørt litt, jeg vet ikke, mitt inntrykk at de kanskje sliter litt med farta. Om det er et reelt problem, eller om det er kanskje noe feil med estimering og forventninger, det vet jeg ikke ennå. Men, jeg vet jo det at for eksempel, at de kanskje sitter og, noe av det jeg prøvde å sniksnakke litt om. Hvordan de skal håndtere problemer. Jeg har jo bare hørt det fra en side, at de kanskje sitter litt lenge og prøver å løse et problem selv for eksempel. Og det er alltid et problem, det er egentlig ikke nødvendig, for du har masse resurser rundt deg å bruke. Det er et sånt typisk utviklerproblem, skal klare alt selv. Andre problemer, jeg ser jo det at de, de har kanskje ikke en ordentlig prosess. Det er ingen på det teamet der som har ordentlig erfaring med

for eksempel Scrum. Så det er kanskje en litt udefinert prosess, kanskje ikke en håndfast prosess. Jeg mener at når en først skal kjøre en prosess, skal en kjøre den håndfast, men løst. Altså det vil tilsi at en kan slippe opp, men gjøre det bevisst. En skal vite hvorfor en gjør det.

Intervjuer: #1 Kunne man gått tidligere og kjørt en hardere prosess på det? #

Scrum-master: #2 Nå kjenner jeg det ikke, men på en generell basis vil jeg si det. Jeg vil si det at skal du begynne med en prosess i et prosjekt, så skal du alltid begynne med, mener jeg av erfaring, så skal en gå inn med et litt sånt firkantet syn. Slik at "dette er prosessen, sånn er den og den skal vi følge." # Ikke sant, og Scrum har noen enkle greie regler, med møter som driver prosessen fremover. Så det er ikke noen komplisert prosess. Og veldig fokusert egentlig. Og å si at en skal følge denne fra a til å, er et veldig godt utgangspunkt. Spesielt hvis en ikke har noen spesiell erfaring fra før. Og så kan man heller ta det jeg kaller "decision-points" eller beslutningspunkter og si det at "Nå er det noe som ikke funker for oss, hvorfor fungerer det ikke?"; "Jo sånn og sånn, vi må kanskje gjøre et avvik." Og da bestemmer en seg for å gjøre et avvik fra den "firkantede boksen" ikke sant. Og etter hvert vil altså prosjektet forme da metodikken til seg. Men det skal være veldig viktig, at man er bevisst på at man tar en beslutning på det man tar et avvik på.

Intervjuer: Så du sier egentlig det at jo mindre erfaring man har med denne type prosess, desto viktigere er det å være strikt på den i starten også?

Scrum-master: Ja, jeg synes det. Det vil jeg absolutt si. Er du mer rutinert, kan du åpne litt mer med en gang, men det betyr ikke at man ha fri sluser. Det vil være mer "OK, nå kjører vi prosessen sånn på dette prosjektet" Ikke sant, og at du da endrer fra basisreglene, for du er fremdeles firkantet, og har fremdeles reglene å forholde deg til ikke sant. Selv om de reglene. Grunnen til at dette går er jo at reglene er nokså løse, men faste nok til å holde formen på prosjektet. Det verste som kan skje egentlig når du bruker metode, er hvis det blir såpass løst, at alle tror du bruker en metode, men du har egentlig ikke en metode. Det er et kjempefaremoment, og det kan fort skje dersom en ikke er bevisst på disse tingene, en gjør avvik fra reglene hele tiden.

Intervjuer: #1 Har du vært borti noen slike prosjekter? #

Scrum-master: #2 Ja det har jeg, og det fungerer veldig dårlig. #

Intervjuer: #1 Hvordan er det, ser du det samme i <Prosjekt 2> også, hvor du innfører metode for noen som ikke har brukt det før? #

Scrum-master: #2 I <Prosjekt 2>, der er de veldig modne for å ta i bruk, de har ikke kjørt ordentlig metode før. De har vel brukt en slags kvasi-fossefall tenker jeg. # Og de bruker veldig, og da mener jeg veldig mye tid på analysefasen. Så på det prosjektet jeg er på nå, så har de brukt mange mange penger på et analysedokument som først har vært veldig stort, og så har de redusert den litt, og så fikk jeg beskjed om at de nå hadde økt den litt igjen. Når jeg da ser hvor greit det går å komme inn der og kjøre brukerhistoriemøte, så ser de virkelig nytten av at dette går lettere, selv om at det at de har gjort en sånn analysedokument på forhånd hjelper litt når du skal gjøre brukerhistoriemøte, da er man forberedt og de vet litt hva de vil ha, men det er veldig sært på <Prosjekt 2> igjen for der vet kunden egentlig (trykk på egentlig) ikke hva de vil ha. Så det er et vanskelig prosjekt, og da vil jeg si det at det er enda viktigere å ha metode, og Scrum fungerer veldig greit. Og brukerhistorier de er så enkle, at det er veldig lett for til og med ikke-teknologer å visualisere litt om hva vi snakker om I tillegg til at du kan få iterasjoner, sånn som første

iterasjon som er definert nå, det er at vi tar de brukerhistoriene som vil erstatte et regneark som de bruker i dag. Rett og slett et Excel regneark. Og de gir de en nokså fokusert, altså da kan vi visualisere nokså greit for kunden. "OK når dere er ferdig med denne iterasjonen kan dere kaste dette regnearket og benytte dere av dette systemet" "I tillegg til det dere hadde fra litt fra regnearket, så vil dere få disse funksjonene i tillegg som addons som gir merverdi." Og vi har hele tiden fokus på verdi. IT-avdelingen der er jo kjempemodne for en sånn prosess, jeg fikk vel det oppdraget da jeg hadde en to timers Scrumpresentasjon for en ledergruppe der. Og kom inn via den veien. De syntes det var kjempebra, har vært interessert i forkant, slik at de kanskje hadde prøvd litt på egenhånd. De har ikke ville innrømme det offisielt, men har ikke fått det til.

Intervjuer: Krever det erfaring fra bruk av Scrum på forhånd før man setter seg inn som Scrum-master? Burde man ha vært i et scrumprosjekt før?

Scrum-master: Det vil jo absolutt være en fordel, og jeg tror det. Hvis en begynner fra scratch, og det har jeg vært med på hvor vi innfører Scrum i en bedrift hvor vi hadde null kompetanse fra utgangspunktet og ikke hadde noen utenfra til å hjelpe oss. Og det er jo mye større kjangs for at du feiler, hvis du ikke har noen erfaring. Og en av de tingene du feiler fort på er at du ikke kan reglene ordentlig. Du har ikke følt scrumprosjektet på kroppen, du har ikke følt reglene på kroppen heller og det gjør det mye vanskeligere å holde linja, for det er klart det at Scrum som en enkelt prosess, den har gitt e ting som en må forholde seg til. Du har morgenmøtet på 15min, du har timeboxet møtene i forkant og det hele. Scrum forventer en veldig strukturert og ryddig arbeid rundt requirements. For å si det sånn. Så hvis du tror at du kan ta Scrum og bare implementere det uten å egentlig jobbe ryddig med definering av produktene dine, noe som egentlig ikke, hvis du ser i mange scrumbøker så har de det som addons, men det er jo ikke en del av selve prosessen. Prosessen sier jo ikke at du skal bruke brukerhistorier på en måte for å definere requirements på ikke sant. Så det er på en måte en slags blindspot til Scrum og så plasserer den ned.

Intervjuer: #1 Er det noen andre prinsipper fra Scrum som du bevisst har valgt å ikke ta med? #

Scrum-master: #2 Jeg må bare tenke på de eksemplene jeg har. Bevisst fra begynnelsen ikke tatt med? #

Intervjuer: #1 Eller bevisst gjort endringer på, som du ikke er enige med at passer til ditt prosjekt? #

Scrum-master: #2 Jeg tror stort sett vi stort sett kjører den nokså etter boka. Men det er klart en gjør noen variasjoner, nå vet ikke jeg helt alt som står. Noen steder står det "sånn" Scrum og andre steder står det "sånn" Scrum # det sier at du for eksempel skal ha en pool med arbeid, ikke sant. så skal teamet bare plukke fra den dynamiske etter hvert som prosjektet går. Det er ikke alltid det går. For eksempel på <Prosjekt 2> så går ikke det, så har de prosjektmedlemmene, de har sine hatter hos <Prosjekt 2>, med spesielle kompetanseområder med veldig liten kryssover, det er litte granne på noen av de, så er det noen av de som er helt isolert. Det går alt fra programmering av pls'r til å utvikle webapplikasjoner. Og da er det sånne avvik, det er mer sånn justeringer egentlig, i forhold til team som blir satt opp. Ellers så følger jeg den prosessen nokså mye. Jeg lar møtene på en måte drive prosessen gjennom. For du har selve iterasjonen. Jeg definerer opp tre møter i forkant. Prosjektet, når du kjører produksjonsdelen, så er det vel fem møter som driver den. Det er de fem Scrummøtene, altså du har: Planleggingsmøte. Da

sier jeg del en er ett møte og del to er ett møte. Så har du morgenmøter og så har du reviewmøte så har du retrospectivemøte. Og så begynner du på nytt igjen. Etter du er ferdig, kjører du prosjektavslutningsmøte. De tre møtene som du kjører i begynnelsen det er kickoff møte, hvor teamet samles og prosjektets mandat og regler blir informert til alle sånn at alle vet hva fundamentet for prosjektet er, brukerhistorier, poengestimering og da lar jeg bare møtene egentlig drive oss gjennom prosessen.

Intervjuer: #1 Sånn som prosjektets regler, er det noe som dere skriver ned som alle har tilgjengelig hele tiden? #

Scrum-master: #2 Sånn som det prosjektet jeg hadde i <Forrige arbeidsgiver>, da hadde vi på en måte, da var jeg ansvarlig for produkt. Og da hadde vi på en måte en håndbok for det prosjektet. Som var skriftlig # Men prosjektregler trenger ikke være skriftlig, det kommer litt an på. Fordi at, mange ting er innlysende, det kommer an på hvor langt du vil strekke deg. Har du korte prosjekter, så vet jeg ikke om det er verdt tiden det tar å sette det ned skriftlig. Hadde Bedriften klart å etablere en mal på det, så er det lettere å ta det skriftlig, men det viktigste er at "baserulsa" er satt, du skal være presis på møtene.

Området til prosjektet er: det er organisert sånn og sånn. Timeføring, det skal gjøres sånn og sånn, kontonr sånn og sånn. Bare sånne elementære ting som er helt på plass. Og i tillegg, så blir det jo, må man jo på kickoff også.. Har man prosjektmanual eller håndbok så skal prosjektets mandat stå der altså. Hva er det egentlig (med trykk) prosjektet skal levere? Bare for å være med og hele tiden holde fokus at man er på riktig vei. Men på de prosjektene i Bedriften, hvor det er veldig kort fartstid, så har vi ikke kjørt noe. Da har vi bare hatt det som det er. Sånn som på <Prosjekt 2> der har vi allerede en bedriftskultur. Som tilsier ting. De har dessverre en litt uheldig bedriftskultur. For de har en kultur som sier at det er greit å, at har du et møte som begynner klokka ni, så er det fem over ni i praksis, ikke sant. Så der har jeg foreslått for de, at de kan kjøre en sparegris løsning hvis de vil på prosjektet. Men siden det er kulturen deres, så er ikke jeg der inne for å reformere deres kultur. Men på morgenmøtene kommer jeg til å være striks, da begynner vi når klokka slår. De som ikke er der, de får litt lett ris på rumpa etterpå, så får vi se hvordan det bærer. Det var start på iterasjon i går, så vi skal ha morgenmøte i dag. Det blir spennende å se hvordan det blir.

Intervjuer: #1 Når er det..? #

Scrum-master: #2 Klokka 10. Vi har ikke sagt tidspunkt, i dag ble det avvik for dette møte. Jeg trodde ikke vi skulle komme i gang med sprinten og iterasjonen i går. # Vi skal ha et fast tidspunkt hver dag. Det kan være at det blir kl ti for å garantere at folk er der. For de er; kulturen er sånn.

Intervjuer: Er det et problem et problem generelt, at kulturen kan være annerledes enn hva metoden tilsier?

Scrum-master: Stort sett, så er folk presise på møter. Det er klart Scrum kommer med en strikt metode, ikke bare det med møtetid. Det er jo normal folkeskikk synes jeg. Og har du kultur som har sklidd ut sånn, og da blir det en sånn, at "sånn er det bare" Folk sitter på kontoret sitt og tenker at "nå er klokka ni, da går jeg bort om fem minutter, vi begynner ikke før da allikevel" det er sånt som bare går sånn ikke sant. Og som påvirker litt. Det som nesten er verst sånn i forhold til kultur det er at mange steder så er det å ha møter en liten kosestund, det er et avbrud fra produksjon og fra jobben sin. Og det har en tendens til at det blir kanskje litt mye tull i møter. Vanskelig å holde fokus for mange. Når du har Scrum med timeboxing på en del av disse møtene, morgenmøtet på

15 minutter, så skal det være litt intenst ikke sant. Samme med disse etablering av Disse to møtene som du kjører i forkant av en iterasjon, for å definere opp iterasjonen. De er også timeboxet, og de er også nokså tichte sånn sett. Selv om du kan slippe litt på, når teamet sitter og definerer oppgaver, så kan du slippe på den, men da begynner det å gå av sprinttiden din. Da begynner du å komprimere den tiden du har på å faktisk levere det du har lovt. Sånn at du kan få litt på en plass, men det blir tatt fra et annet sted, ikke sant. Så da er det litt viktig at du holder fokus. Og det kan være litt vanskelig av og til, men igjen, ofte hvis som hos <Prosjekt 2>, når vi kommer inn, de er så positive, de ønsker å få inn metode og de ønsker å ha Scrum, de er veldig positive til det. Det er klart når du har et sånt positivt utgangspunkt, så er det lettere å holde fokus på folk og dra de med deg gjennom. Enn hvis du har, en bedrift hvor folk kanskje ikke er helt motivert. Men den casen har jeg ikke vært borti ennå, så den kan jeg ikke uttale meg om. Hvor det har vært folk som "her kommer du... dum", ikke sant, "skal gjøre noe fancy"

Intervjuer: #1 Det er vel typisk en kan se i kommunen og... #

Scrum-master: #2 De skal jobbe på sitt vis. Og det tar jo litt av rollene, for eksempel.

Noen steder så kan det godt hende en prosjektleder kan føle seg litt truet, siden prosjektlederrollen egentlig er ut. #

Intervjuer: Har du lyst til å forklare litt om din rolle som Scrum-master, hva er dine arbeidsoppgaver?

Scrum-master: Hva mine arbeidsoppgaver er. Jeg begynner bare å nøste i en ende, så kan vi se hvor vi ender opp. En ting er i alle fall å hjelpe med prosessen til prosjektet Og sånn som jeg ser det da. I alle fall sånn som vi kjører litt. Et prosjekt har på noen måte bruk for en prosjektleder. Selv om han ikke driver prosessen. Så det blir litt prosjektlederrolle i tillegg. Men det er ikke mye, det er mer det der at "Du er Scrum-master på det prosjektet, javel hva betyr det?" "Jo, da har jeg tenkt å samle sammen trådene." Så da kjører vi kickoffmøte for å kjøre det i gang. Så får vi på plass reglene, og så er det egentlig bare å fasilitere prosess. Altså, hvor jeg da ser for meg en Scrum-master sånn som jeg lever den, det er et prosjektmedlem, som enten kan være med i produksjon eller ikke, som bare kommer inn som Scrum-master på prosjektet. Prosjektmedlem som bare skal være med i prosessen med den hatten han har. Den hatten er i utgangspunktet ikke sjef, men medlem og han skal bare sørge for at prosessen er på plass, prosessen holder og at man ved hjelp av prosessen drar seg gjennom prosjektet. Og at hvis det er avvik, kunne begrunne det. Samtidig så hvis det skal være nødvendig, så må han også kunne stå opp som en leder, og si "Ok, jeg er Scrum-master, nå må jeg bestemme dette" Sånn er det bare. For jeg mener og at, i tillegg til at. Sånn som når. Og det kan du si det at du skal justere prosessen, av og til må du ta den rollen, jeg liker ikke å ta den rollen, for jeg liker best hvis teamet selv kan sette retningen ikke sant. Men, for det er jo det som og er tanken. Men av og til må man også sette opp et styre. Så må man passe på hele tiden at siktet er riktig, og passe på at kula følger den rette banen, i forhold til våre mål. Så det er egentlig det jeg ser på som Scrum-master. Tar de administrative tinga, bare etablere og sørge for høyest mulig produksjon. Selvfølgelig under prosessen, er det noen hindringer så skal jeg rydde de av veien.

Intervjuer: #1 Hva er et typisk hinder i <Prosjekt 2>. #

Scrum-master: #2 I <Prosjekt 2>? Nå har vi ikke kommet sånn ordentlig i gang der ennå, men hvis jeg skal se hittil. Et typisk <Prosjekt 2> hinder, det er at de ikke har, at det er en litt uproff organisasjon. # Altså det vil tilsi at de har, <Prosjekt 2 - underavdeling> er et

konsulentselskap i <Prosjekt 2> på en måte. De leverer IT og datatjenester utvikling, og da er det det at de ikke har ordentlig system på tingene. I dette problemet her så har de et produkt de skal bruke, en komponent komponent/produkt som de skal bruke internt. Det er ingen produkteier på det Så hvem som egentlig sitter på den kunnskapen, jo, det er nå en person som var ansatt i <Prosjekt 2> for ti år siden, som nå er innleid, og problemet er at de vil helst bruke den, fordi det er <Prosjekt 2> standard, men de kan ikke bruke den rett ut av boksen. Det kan være at de bare kan bruke noen felter til en annen funksjon, et sånt avvik, eller så må vi endre produktet, eller så må vi forkaste det og lage noe nytt. Og det er et problem. Det er vanskelig å kunne ta den beslutningen, hvor kan det gjøres? Det er ingen som sitter på den ordentlige kunnskapen på det produktet. Det er kanskje ikke noe direkte, det blir en hindring i prosjektet, for det er definert opp at arkitekturen er sånn, så nå begynner det å dukke opp at det ikke er sikkert at vi kan bruke den ut av boksen allikevel. Det begynte i går når vi satt oss ned og la aktivitetene ut.

Intervjuer: #1 Hva gjør du da som Scrum-master? Må du ta den beslutningen da? #

Scrum-master: #2 Jeg kan ikke ta den beslutningen som Scrum-master. # Hadde jeg vært i <Prosjekt 2>, fast ansatt, så kunne jeg kanskje tatt den beslutningen. Jeg må bare sørge for at de folka som kan ta den beslutningen tar den, og det er min jobb i dag. Når jeg går hjem i dag, så må den være tatt. For at hvis ikke, så kommer han ene, om to tre dager sitte ledig og bare håpe på at den beslutningen blir gjort. Så den må gjøres nå Og det er veldig viktig at en gjør den riktige beslutningen også. Så det blir spennende. Og det er en sånn hindring som dukker opp. Andre hindringer, hos <Prosjekt 2>, der har jeg ikke noe jeg egentlig skulle sagt. Hvis jeg tenker tilbake til <Forrige arbeidsgiver>; jo andre hindringer. Helt klart, det er det hvis det som kalles kyllinger blander seg inn midt i en prosess, det er et problem.

Intervjuer: #1 Kyllinger? #

Scrum-master: #2 Ja du kjenner ikke til det? Det er en god historie på det som du kan bruke som huskeregel. # Høna som spurte grisen om han skulle være med å starte restaurant. Grisen tenkte seg godt om og svarte at det kunne han tenke seg. Hva skal den hete? spurte grisen. Da svarer høna: bacon and eggs. Da svarer grisen: Jeg tror ikke jeg skal være med på dette allikevel. Høna kan bare legge egg på løpende bånd, men for å få bacon er grisen mye mer involvert. Grisene er altså mye mer involvert i prosjektet, Det er da teamet, Scrum-master og produkteier. Kyllinger er alle de som har interesse av produktet. For de kan også være med på gitte ting i en Scrumprosess. Det som er et problem, det er hvis kyllinger ikke klarer å holde seg nøytrale i løpet av en iterasjon. At de blander seg inn. Så blir det litt verre.

Intervjuer: #1 Så det er et typisk problem, at "utenforstående blander seg i prosessen? #

Scrum-master: #2 Ja det kan være. Nå snakker jeg bare med min erfaring fra <Forrige arbeidsgiver>. Jeg har ikke den lange fartstiden nå hos Bedriften til å komme så langt i Bedriften sine prosjekter. Og der har du kortere prosjekt med kortere forhold til kunde som regel. Så det blir spennende å se hvordan det blir. # Men, og det er et problem hvis sånne aktører går rett til prosjektdeltagere og det er for eksempel autoritære personer i en organisasjon. Som sier at en bare skal gjøre noe. Da er det ofte at sånne personer bare slipper det de har og så gjør de det. I Scrum så skal, som Scrum-master så skal jeg vite om det. Så skal jeg bestemme om de får lov til å ha noen av resursene eller ikke. Ikke sant, for det er en viktig ting som Scrum-master, så skal jeg passe på prosessen, at teamet committer seg. Om en blir tatt ut av teamet en liten stund så går det ut over

produktiviteten. Om det bare er en time så koster det en halv dag, bare i tap av fokus og andre ting. Og, så det er i alle fall en ting. Andre hindringer, kan være elementære ting. En harddiskkrasj for eksempel. Det er min jobb å fremskaffe, bruke den myndigheten en har til å fremskaffe en pc så fort som mulig. Eller få den fikset så fort som mulig. En annen ting kan være hvis noen blir syke, at du da og da må man se litt på det, hvis man har noen som er syke og en ser det at her er det litt mer enn bare en dag, hvis det går tre fire eller en uke eller to, hvis det er noen som er alvorlig syke, så påvirker jo det hele iterasjonen og da blir man nødt til å gjøre noe drastisk. Fordi at da, da vet man allerede da at en ikke har sams til å levere alt. Og da er det viktig at man i alle fall leverer hele ting og ikke mange halve. Det er klart at hvis man har syv brukerhistorier, er det fort lurt, at når man kommer til endinga, så har man to ferdige, så er det fem som er halvveis. Så da må man jo passe på det, men det er jo en hindring egentlig i prosjektets gang. Tenker på andre hindringer jeg hadde der nede. Jeg har ikke noen komplett smørbrøddliste i hodet liksom, jeg bare tenker tilbake til erfaringer og se hva jeg husker.

Intervjuer: Tilbake til kyllingene som involverer seg, hvordan løser en en slik case hvor det er en autoritær person som kommer inn og endrer?

Scrum-master: Man kan aldri hindre at disse kommer inn ordentlig, for det er opp til de å gjøre det. Men det er klart at en er nødt til å oppdra, skal du ha Scrum i en organisasjon så må organisasjonen vite hva det er. Informasjon er veldig viktig. At folk vet hva scrumprosjekt er og er litt informert om reglene. Kan ikke vite detaljer, men hvordan de og prosjektet skal spille sammen ikke sant Det er jo klart også en fordel, du bør jo ha det, du bør jo ikke innføre noen metoder i en organisasjon uten at du har noen høvdinger med for å si det sånn Det gjør det i alle fall vanskelig. Da føler de at de mister kontrollen, og du har ikke deres backing og når ting begynner å skorte på og du trenger deres backing, så har du ikke det. Så du kan si det at sånne kyllinger som er autoritære de kan komme inn fra to nivåer. De kan komme fra rett overstående overordnede, av prosjektmedlemmene og Scrum-master og du kan ha folk utenfra som kommer. Altså fra andre deler, sidehirearkier som kommer inn. Av de som er rett ovenfor, da er det vanskelig, men på mange måter enkelt allikevel. Så lenge du har deres commitment med når er innført. Da er det bare å si at "Det kan vi ikke gjøre, sånn er det bare" Hvis vi skal gjøre skal ha en ut to timer, så må du begynne å forklare konsekvenser, og da må du la han ta avgjørelsen, på mange måter. For det som er vanskelig i akkurat de casene, er at du har en organisasjon der du skal performe ovenfor kunder. Avhengig av hvor stor organisasjon du er, så sitter kanskje en del nøkkelresurser på teamet ditt, Som kanskje må brukes til andre ting. Så en av de viktigste tingene du gjør rundt det, er at passe på at hvis du har en nøkkelresurs i en organisasjon som må, som du vet at han må av og til gjøre noen andre ting, skal du aldri kalkulere med han 100%. Da må man være åpen på det at da er han deltid i prosjektet. For å kunne håndtere andre ting. Det er klart at hvis en glemmer det, eller finner ut at en har en sånn nøkkelperson i prosjektet, så må en jo finne ut av hvordan en skal løse problemet. Kan jo vente til neste iterasjon, og så planlegger man, begynner man å planlegge til da, så må man begynne allerede nå. Sånn at, men nå begynner vi å avvike til prosessen igjen ikke sant. Og da er det viktig igjen, at en har beslutningspunkter. Jeg er av den typen at en skal være strikt, men igjen sier jeg det med avvikene. Avvikene kan være av mange forskjellige typer ikke sant. Og da må man veie det opp mot, sånn som når jeg jobbet i <Forrige arbeidsgiver>, leverer produkter som de produserer, og da er det klart at kunden er det viktigste, for uten kunden så har vi ikke

noe business. Det jeg så der, var at i begynnelsen så var det et helvete for å si det mildt. For vi ble forstyrret hele tiden Men etter, og det kan du si mye etter bedriftens modenhet også. Vi hadde vært et grunderselskap som begynte å utvikle produktet vi skulle selge og det du ofte har da det er at du har ikke moden nok produkt til å bare gå og installere, du er nødt til å bygge funksjonalitet for hver kunde du legger inn så vokser produktet ditt sånn sakte men sikkert. Til slutt så kommer du til et brekkpunkt, hvor du kan si: "Ok produktet vårt er godt nok til at vi kan levere mye verdi til 80 % av kundene våre. Nå må vi begynne å snu på flisa" Frem til da så er du litt reaktiv, og da kan du si at det ikke er sikkert at Scrum er den beste prosessen heller. Du får en jobb. Vi må ha den funksjonaliteten fort som F. ikke sant. Men må komme over den grensa og snu på flisa, så er det veldig viktig at du, for å oppdra organisasjonen med det hele. Og det tok litt tid, men nå begynner det å snu. Og det du, det som vi så, de grepene som vi måtte gjøre for å snu hele flisa det måtte bare være håndfaste. Altså når vi begynte iterasjonsplanlegging, måtte vi få med de aktørene som faktisk var størst uromoment med i den prosessen. Og si at " Nå låser vi visse resurser sånn, vi skal utvikle produktene sånn og sånn" Og så forklarer de konsekvensene at da skal de være låst. Og så når du får commitment på det så går det kanskje en eller to iterasjoner hvor du får litte granne, men etter hvert så vi det i praksis at det begynte å gli litte grann vi får snudd litt på flisa. Selv om man fremdeles har litt ad-hoc jobber for eksempel bugfixing, som kanskje man har et team med et produkt må ta. Men da må man bare si at scrumprosessen tar så og så mye tid så har vi så litt tid tilbake som går på å vedlikeholde de andre produktene hvis det kommer noen kritiske som må fikses og shippes til kunde.

Intervjuer: #1 Dere legger rett og slett en brukerhistorie på bugfixing i hver iterasjon? # Scrum-master: #2 Nei jeg ser det ikke som en del av iterasjonen, men at du fristiller resurser. At du kalkulerer ikke med 100 %. # Og det er egentlig to produkter. For det du skal levere ut i fra Scrum iterasjonen, det er på en måte en nytt produkt, eller en ny versjon, men du fikser på den gamle.

Intervjuer: #1 Sånn er det vel i andre pågående prosjekter.. #

Scrum-master: #2 Ja det er akkurat det, så det er jo alltid det. Men jeg tror i alle fall det viktigste, når du har de overordnede som er rett ovenfor deg, at de blir involvert og committa, at de endrer tankesett. For det at igjen Scrum, det er veldig mange som for eksempel synes det er vanskelig å forholde seg til Scrum. Men vi har opplevd i praksis så endrer de litt på meningen sin allikevel. # Og spesielt hvis du ser det at man har den lille fleksibiliteten. OK vi kalkulerer ikke med de 100 % med de resursene som er viktige andre steder. Men vi skal ha de den prosentandelen som vi har sagt. Eksterne aktører, det er jo vanskelig, men lettere allikevel. For det at ofte kan du bre si nei, for du har ikke noe ansvar til de. Hvis de blir et problem kan du eskalere det opp. Og si det at "Jeg har fått dette mandatet, og det er min jobb" Du får gå og snakke med noen overordnede. Hvis du da har kommet ut der og de har begynt å bli oppdradd, så løser det seg ganske greit. I <Forrige arbeidsgiver>; er det noe som er skikkelig kritisk, så er det jo klart at en kan si at en avslutter iterasjonen. "Og så trår vi til, og så oppretter vi bare en ny iterasjon "down the road" når vi er på plass igjen" Vi gjorde aldri det i <Forrige arbeidsgiver>, der var vi nok ikke så strikte, der hadde vi et problem om at de var litt for slepphendte Der hadde vi nok at vi, der endret vi nok litt på iterasjonen underveis. Og, det er noe jeg synes er fyfy. Etter hvert så gikk det seg noe bedre til da, men det tok litt for lang tid. I <Forrige arbeidsgiver> så tok det litt for lang tid fra vi begynte å innføre prosessen til den gikk

strømlijeformet. Et av de grepene vi tok, det var at vi faktisk måtte stramme inn på disse tøyene. Det er der jeg har det med å heller begynne strikt og heller slippe ut, enn å gjøre det omvendt fra. For det har jeg gjort og det fungerer ikke. Det tar lang tid før du får ting på plass.

Intervjuer: I forhold til <Prosjekt 1>. Scrum har en ganske strikt backlog, med det som er høyest prioritert øverst hvor teamet skal velge de høyest prioriterte oppgavene først. Det er ikke noe som blir fulgt hver gang i det prosjektet?

Scrum-master: Det som er med den backloggen, den er egentlig dynamisk. Det som er, er at når du begynner å planlegge en iterasjon så har du backloggen. Og så, er det basert på den og prioritert det som er viktigst. Nå sier vi teoretisk ikke sant. Det som er viktigst er det som skal inn i iterasjonen. Når vi så kommer til neste iterasjon, kan ting ha endret seg. Det vil tilsi at det kan ha blitt tatt ut noen brukerhistorier fra backloggen, det kan være lagt inn noen nye. Noe kan ha endret seg for den saks skyld. Prioriteten kan være endret på tingene. Men allikevel, resultatet av disse endringene er utgangspunktet for å planlegge neste iterasjon. Og så er det jo klart at normale avvik som du gjør fra. Og grunnen til at, prioriterer du backloggen, så er det jo klart at du skal implementere det som har høyest verdi til enhver tid og det er det som er prioritert høyest. Det er nokså naturlig. Så kan du gjøre avvik fra det, det kan være i forbindelse med det at det kan være avhengighet mellom brukerhistorier. Det kan være det at teamet ser det at "hei, det var et par av disse her som var litt for store, vi klarer ikke alle disse store" Det kan godt hende vi kan droppe en av de som har høyere prioritet, men så får du en eller to under der som er mindre. For å klare å matche størrelsen på iterasjonen. Jeg har og vært med i avvik, hvor du for eksempel selv om du har nok prioritet 1 brukerhistorier, har endt med å plukke en oppgave 2 for eksempel. I den greia. Og min konklusjon av det, det er det at enten, fra man prioriterte backloggen sist gang til man begynner å plukke ut så er det gjort endringer i oppgaver rundt som gjør at en har andre verdier og prioriteringer eller b, man har rett og slett prioritert feil. At det da, "Den toeren skulle egentlig være en ener den."

Intervjuer: I <Prosjekt 2>, der var det problemer med produkteier, eller har dere en produkteier som har ansvar for backloggen?

Scrum-master: Ja; og nei. Vi har en produkteier og etter i prosessen, skal han ha ansvar for den backloggen. Det har han ikke forutsetninger til å kunne gjøre i det hele tatt. Og med tanke på at han er bare produkteier for dette lille prosjektet, så er han ute, så kan det godt hende det går to år at han aldri vil være i den rollen igjen. Så er det ikke hensiktsmessig å trene han opp for å kunne ta fullt eierskap på den eller det vil si administrasjon. Der har jeg som Scrum-master og ho som egentlig er da på scrummaster-opplæring fra <Prosjekt 2> sin side, vi har tatt på oss den praktiske rollen. Men det er produkteier som hele tiden tar beslutninger. Vi tar oss av selve skrivejobben og administrerer den, men listen med brukerhistorier og alt, det er synlig for han. Men det er vi som redigerer den. Men vi passer hele tiden på, for det er jo en vanskelig situasjon. Men vi passer hele tiden på at han er klar over konsekvensene, det er alltid han som skal ta beslutningene på det nivået hvilke brukerhistorier som har prioritet. Vi kan stille spørsmål, for å prøve å opplyse han om hva han egentlig har bestemt seg om. For det at, litt må de lære seg, de må få en gitt oversikt på tingene. Og det tror jeg vi har klart. Men han er allikevel ikke moden nok for å kunne fysisk si det at "her er den backloggen, si i fra nå kan du gjøre hva du vil med den til neste iterasjon, så setter vi oss ned og ser hva

du har gjort da". Og mye av det går også på at han vet ikke helt hva han vil bestille. Altså <Prosjekt 2> Research, IT-avdelingen som har jobbet med masse prosjekter der, de vet mer hva som trengs, uten at de vet alt. De må ha løst noe, men jeg merker det, vi må stille en del ledende spørsmål til de. For det at, hvis de er basert på andre løsninger, så ok "er dere interessert i det; ja" Og så kan man begynne å nøste seg. Men jeg synes det at, for eksempel på workshopen for å definere brukerhistorier gikk veldig bra. Det var tamt de første 15minuttene, men det er helt typisk når en kjører en workshop med brukerhistorier hvor det er folk som ikke har vært med på det før. Men når folk begynner å bli varme i trøya og ser konseptet, altså det går bare på to ting. Skjønne den setningen som skal skrives, hvorfor man skal gjøre det. Og bare komme inn i den modusen sånn at disse setningene klarer du å visualisere litt i hodet. Det er det jeg prøver å innprente. Det er bare å sette brukerstolen og se for deg hva du skal gjøre med datamaskinen når du ha denne her applikasjonen ikke sant, så vi formulerer dette med disse setningene. Og da går det, jeg føler likevel at han har et godt eierforhold til det som er, vi klarte også å prioritere det på det han, så har han en under seg som er med, han er produkteier så har han en med seg som er kylling da på en måte, men som er med og bidrar i disse prosessene der. Og, da synes jeg det fungerte ganske godt, de har klart å komme opp med de meste brukerhistoriene ved hjelp av at vi som team da var med å fasilitere litt og hjalp de litt på vei, men kom allikevel frem med grunntankene. Sånn som da, der, det er mange måter å begynne et sånt møte på. Men da begynte jeg se meg for et business case, ikke sant, for de visste ikke helt hva de skulle ha. De visste bare at de hadde et regneark, og det skulle erstattes. Og så skal de gå litt videre i prosessen, de er jo ikke kyndige i det hele tatt på det nivået. Å begynne med et business case hvor jeg definerte "Hvorfor har dere regnearket i dag?" Og alt baserte seg tilbake igjen til det med sporing og kvalitetssikring i ettertid, altså sporing i forhold til det å kunne se hvis noen klagde på kvaliteten på produktet så kunne du se at "Ja, her hadde vi avvik, det stemmer at det var et dårlig produkt." Og eventuelt kunne få en erfaringsbase til kunden og kunne si at "hvorfor fungerer prosessen sånn som den gjør" Men, hovedspørsmålet, nå fløy vi langt ut her, men hovedspørsmålet var jo Om vi tar backlog, jeg vil si ja, men med sterk veiledning ved hjelp fra Scrum-master. Jeg tror nok de har følelsen at det er de som er produkteier og at det er de som bestemmer retningen og det er de som tar beslutningene på det nivået. Intervjuer: Kan man si det på generell basis, at det er en utfordring å få de som er prosjekteiere til å forstå prosessen?

Scrum-master: Nei, jeg vil ikke si det på generelt grunnlag. Du kan tenke deg <Prosjekt 2> er en industribedrift, du kan tenke deg de vi forholder oss til er egentlig ikke IT-mennekser. IT er kanskje ikke førstehånd til de. Sånn at de har en litt annen bakgrunn, selv om de er ingeniører, så er de en annen type ingeniører. Som gjør at de har ikke den IT-interessen, de er veldig opptatt av prosessene det som skjer der ute på anleggene. Sånn at det er en annen. Det går nesten på mennesketypen og da kan du si; og interessene deres Sånn som i <Prosjekt 4>, der tror jeg det kommer til å gå kjempegreit. Der har vi en ung mann som er produkteier. Og, han var vel ferdig å studere til sommeren tror jeg. Så han er bevandret og kan bruke en datamaskin. Han har tenkt mye på dette her og har mange ideer. Så der tror jeg det kommer til å gå mye greiere. Det prosjektet der tror jeg kan, når jeg har hjulpet han å sette opp den backloggen og få alt det på plass, så tror jeg at en kan gå på det nivået at en kan si "Nå, nå er det du som styrer den." Men, jeg ville allikevel ha informert om alle endringer. Jeg vil kreve at når vi skal begynne å planlegge

neste iterasjon så skal han fortelle meg hvilke brukerhistorier han har endret hvilke han har lagt til og hvilke han har tatt fra. Det er ikke fordi jeg trenger kontroll, men det er for å ta vare på det kommersielle aspektet. Sånn at hvis det er endringer som vil påvirke prisinga av produktet vi leverer til de, så må jeg vite det sånn at vi kan si det til de og si: "Ok, det er greit, men det kommer til å påvirke prisen". Men ellers, så ville jeg ikke ha behov for det. For da er det eneste jeg trenger å vite, det er at "nå skal vi planlegge neste iterasjon, hva har vi som utgangspunkt?" Og da har vi brukerhistorier, som er prioritert. Kjempegreit.

Intervjuer: #1 Endringsforslag kommer da til å komme til han, og ikke deg? #

Scrum-master: #2 Ja, det tror jeg nok, for det at nå er det jo han som hovedsaklig kommer til å styre den produktutviklinga der, men allikevel han kommer til å sitte hos de, så det ikke så mye vi kommer til å ha til de andre. Det er han vi hovedsaklig kommer til å ha et forhold til. Kanskje de andre når vi skal ha disse produkt review møter og det ikke sant, men akkurat det tror jeg absolutt vil gå. Når vi ruller ut en versjon på en testserver som sikkert han kan leke med, mens vi utvikler andre iterasjon for eksempel og viser det til de andre der nede. Er det han de kommer til å komme med forslagene til. # Jeg regner med at i begynnelsen så vil han kanskje ha hjelp av meg til å innarbeide de forslagene i produktbackloggen, jeg tror det skal gå veldig fort. Selv om det er et veldig kort prosjekt. Intensivt, to måneder. Det skal leveres 1. mai.

Intervjuer: #1 Er det to iterasjoner, eller er det? #

Scrum-master: #2 Jeg tror vi tar to ukers iterasjoner, så vi får sånn sett fire. At, nå skal vi ha brukerhistorieworkshop på torsdag. # Så må vi se hvor langt vi kommer da. Jeg hadde håpet vi kunne begynne på iterasjonen til mandag. Men det er ikke sikkert vi klarer. Det forutsetter at brukerhistoriene er på plass, prioritert og poengestimert.

Intervjuer: Til det med iterasjonslengde, hva er kriteriene for å sette lengden på de?

Scrum-master: hehe, magefølelsen. Nei der. Jeg liker å si det slik at en må se på egenskapene til en organisasjon, eller egenskapene til prosjektet. For eksempel, for å ta det eksempelet fra <Forrige arbeidsgiver>, der hadde vi jo lange iterasjoner. Opp til seks uker, men da hadde vi fire uker med utvikling og de to siste ukene innen iterasjonen, det var kun QA og det å produksjonssette. Det vil si at etter de seks ukene skulle du ha et nytt produkt som kunne shippes rett ut til kunde. Alltid. Og du kan si det at når du har så lange så er det mer, så er du nesten mer garantert å kunne få forstyrrelser fra eksterne parter. Har du kortere så kan du si at en fikser det etter den pågående iterasjonen. Så det er jo klart. Det er en ting man kan tenke på. En annen ting en kan tenke på, det er jo det også for eksempel når en nå har to måneder å lage et produkt på, da synes jeg ikke det er helt fornuftig å ha fire ukers iterasjoner heller. Da rekker jeg så vidt to iterasjoner. Og da føler jeg ikke en får den styringa underveis, som kanskje er interessant. Men jeg synes en ukes iterasjoner i utgangspunktet er for kort. Det mener jeg skal være sære tilfeller. Og det er nok kanskje et problem hos <Prosjekt 1>, det vet jeg ikke. Så det er noe jeg vil se på etter hvert, når jeg blir litt mer kjent med prosjektet og det jeg har, men det kan være det at de har for korte iterasjoner

Intervjuer: #1 Det kan bli litt mye overhead #

Scrum-master: #2 Ja ikke sant. Sånn at en ukes iterasjoner, det er et unntak, som kan være at man at man av og til kan kjøre. Men jeg tror nok ikke det er uten grunn at en sier at iterasjonslengde er to til fire uker. # Stort sett hvis du leser. Det er noen som sier fire uker. De fleste sier to til fire uker, i alle fall det jeg har lest. Det er nok en grunn til det.

Men jeg synes og det er greit å ha muligheten til å si at vi kjører en ukes iterasjoner. Men du skal likevel da tenke litt på det at du tar en del av disse utgangspunktene. Så sier de for eksempel at, har du 30 dagers iterasjon, så har du timeboxet for eksempel disse to møtene for å definere iterasjonen ikke sant gjennom produkteier og det med teamet hvor du setter oppgavene. De er timeboxet til å være fire timer ikke sant. Og går du ned til en uke for eksempel, så må også sånne møter skaleres ned. Jeg tror aldri du klarer å skalere ned de møtene til en time per. Men du bør i alle fall si at du ikke skal gå over to timer på de. I alle fall ikke. Men da kan man si at du allerede der har dobbel overhead. Ikke sant, og når du først har et reviewmøte, teoretisk sett, skal da også reviewmøtet bli kortere fordi at du har bare halvparten av funksjonaliteten du har utviklet på en uke i forhold til to uker. Men allikevel, det er alltid en del overhead med sånne møter også, så det å presentere på to uker, istedenfor en uke har heller ikke så mye å si. Det samme med reviewmøte, når du først har et reviewmøte så du har en eller to uker med erfaring. Så har ikke det så mye å si, det er mer på... Ja det tror jeg sånn at en ukes, jeg kan ikke si jeg har noen caser hvor jeg synes det er riktig. To uker synes jeg er veldig, synes jeg er maks(minimum). På <Prosjekt 2> så kjører vi tre uker.

Intervjuer: #1 Noen spesiell begrunnelse for det? #

Scrum-master: #2 Det det gikk på der, var at de hadde fire uker, og noe av det gikk rett og slett på kostnad. De har bare fått penger for å kjøre en iterasjon av kunden sin. # Og da ble det litt i forhold til hvor mye penger som var tilgjengelig. At da hadde vi definert fire uker. Så hadde han kanskje steilet litt på pengene. Så det var rett og slett en økonomisk begrunnelse. Han hadde kanskje blitt mer happy med, hvis vi hadde tatt det på to uker, for da hadde det blitt enda mindre penger som hadde gått ut, men du skal kunne levere noe og der er det.

Scrum-master: Sånn som på <Prosjekt 2>, så kjørte jeg introduksjonsrunder på alle som skulle være med. Og sånn minikurs, for å gi den selve forståelsen, også vil en jo lære etter hvert og se hvor hullene er, men det viktigste er at folk kan se prosessen fra a til å overhode sett. Alle, de som er med, produkteier og de. Og hva sjekkpunktene er underveis. Så hvordan man utfører de tingene i mellom, det kan man jo se på, men den kommunikasjonen er jo kjempeviktigviktig. Sånn som med <Prosjekt 4> så har jeg hatt et to timers møte nå, hvor vi; ikke prosjektkickoff for vi har ikke alle medlemmene ennå. Du kan si et kickoff mellom Bedriften og <Prosjekt 4> da, for å si at "Nå skal vi begynne" Hvor jeg bare fortalte han mer detaljert prosessen hvordan vi jobber fra a til å. Disse møtene, hva vi går gjennom Og våre forventinger til han som produkteier, hvordan er involveringa hans, mye i begynnelsen, synker litte grann, og så blir det mye på slutten i avslutninga kanskje. Per iterasjon, ikke sant. Disse fasene som er i forkant og i etterkant. Altså prosjektoppstart og avslutning. Så han kan se hele gangen i det. For det verste er, når du begynner å ha folk som føler at de roter i mørket og er totalt avhengige av at du leder de gjennom prosessen som en blind mann. Det betyr at de ikke kan tenke selv. Da har de ikke skjønt det, hvis det hadde dukket opp noe, da hadde de ikke skjønt hva de skulle gjøre for noe.