

Prosjektstyring

Luxmi Pal & Ka Way Lay



Forord

Denne oppgaven er skrevet som en avsluttende obligatorisk innlevering av masterprogrammet i Økonomi og Administrasjon ved Universitetet i Agder, under spesialiseringen Økonomisk styring og prosjektledelse.

Arbeidet med denne oppgaven har bydd på en rekke utfordringer, da vi begge var i fulltidsarbeid til å begynne med. Vi har lært mye om oss selv og av samarbeidet vårt. Endelig er vi ferdige!

Først og fremst vil vi rette vår store takknemlighet til vår kjære veileder, eller ”villeder” som han likte å kalle det, Øystein Husefest Meland. Tusen takk for at du ga oss inspirasjon, veiledning og presise konstruktive tilbakemeldinger under arbeidet av denne oppgaven. Vi hadde ikke klart det uten deg. Du har åpnet en ny og interessant verden for oss innen prosjektledelse.

En stor takk rettes til Red Rock As og deres ansatte for hjelpen til våres forskning og intervjuer. Vi vil også takke kunderepresentanten fra Norway Bussekspress As som tok seg tid til å la seg intervjuer. Spesielt vil vi takke Morten Rønning fra Red Rock As for din endeløse tålmodighet til vårt konstante søk etter informasjon og data. Du har vært til utrolig stor hjelp. Vi håper denne oppgaven kan være et bidrag til deres bedrift.

Tilslutt vil vi takke venner og familie som har støttet oss gjennom denne prosessen. Spesielt vil vi takke Afshin Fiaz, Stina Andersen og Silje Brandsvoll for korrekturlesing av oppgaven vår.

Kristiansand, desember 2013.

Luxmi Pal

Ka Way Lay

Sammendrag

I de siste årene har det vært populært å ta i bruk Scrum-metodikken som er en av de agile tilnærmingene i prosjektledelse av IT-prosjekter. Hensikten med denne oppgaven har vært å finne ut om Scrum tilnærmingen fungerer som en effektiv styringsmetode. Problemstillingen ble som følgende:

«Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?»

Oppgaven begynner med generell teori om prosjekt og prosjektledelse, deretter sammenlignes teorien av tradisjonelle og agile metoder. Det blir presentert diverse styringsverktøy som benyttes i disse metodene.

Vi valgte å følge et prosjekt i bedriften Red Rock. Kvalitativ tilnærming ble benyttet og data ble samlet inn via intervju av nøkkelpersoner i prosjektutviklingsprosessen. Videre ble dataene analysert og drøftet. Til slutt ble det gitt forslag til forbedringer.

Resultatet ble at det ble funnet en del god mangler ved Scrum som styringsmetode, da metoden anses som en design metodikk. Det fantes nesten ikke kontroll og oppfølging. For å effektivisere styringen anbefales det å benytte seg av tradisjonelle styringsverktøy for å få mer kontroll.

Innhold

Innhold	III
Kapittel 1 - Innledning	1
1.1 Mål og motivasjon for oppgaven	1
1.2 Red Rock Gruppen	1
1.3 Problemstilling	3
1.4 Problemavgrensning.....	3
Kapittel 2 - Teori	4
2.1 Prosjekt.....	4
2.1.1 Prosjektmandat	6
2.2 Prosjektstyring.....	7
2.2.1 Styringsnivåer og former.....	8
2.2.2 Styringsfunksjoner og -faktorer	9
2.2.3 Fremdriftsplan	10
2.2.4 Endringshåndtering.....	11
2.3 Styringsmodeller	12
2.3.1 Tradisjonelle metoder.....	12
2.3.2 Agile metoder.....	14
2.4 Planlegging i tradisjonelle modeller	21
2.4.1 Prosjektnedbrytning (WBS)	21
2.4.2 Milepælmetoden – overordnet nivå	22
2.4.3 Critical Path Method (CPM)	23
2.4.4 Programme evaluation and review technique (PERT).....	25
2.4.5 Gantt – diagram.....	27
2.4.6 Kostnadsplanlegging	28
2.4.7 Ressursplanlegging	32
2.5 Planlegging i Agile modeller	34
2.5.1 Dekomponering av produktkrav	34
2.5.2 Releaseplan.....	36
2.5.3 Iterasjon plan.....	39
2.6 Oppfølging i tradisjonelle metoder.....	42

2.6.1 S-kurver - Integrert KTR-styring	43
2.7 Oppfølging i Agile Metoder	47
2.7.1 Release burndown.....	47
2.7.2 Velocity	49
Kapittel 3 – Metode	50
3.1 Innledning	50
3.2 Kvantitative og kvalitative metoder.....	50
3.3 Valg av metode	52
3.3.1 Intervju	52
3.4 Datainnsamling	54
3.5 Dataanalyse.....	54
3.5.1 Validitet og reliabilitet	56
Kapittel 4 – Empirisk data	58
4.1 Referanseprosjektet Mobile Ticket	59
4.2 Intervju med Redrock.....	60
4.3 Intervju med Flybuss Ekspress AS.....	68
Kapittel 5 - Analyse	72
5.1 Tradisjonelle VS Agile.....	72
5.1.1 Forskjeller.....	72
5.1.2 Likheter	74
5.1.3 Fremdriftsstyring	75
5.2 Resultater og drøfting	78
5.2.1 Styringsmodell.....	78
5.2.2 Fremdriftsplan	81
5.2.3 Oppfølging.....	84
5.3 Red Rocks forbedringspotensial.....	88
5.3.1 Forbedringsfaktorer	88
Kapittel 6 - Konklusjon	93
Kapittel 7 – Fremtidig forskning	95
7.1 Hybrid metode.....	95
Referanseliste	96
Vedlegg.....	98
Vedlegg I	98

Vedlegg II.....	102
Vedlegg III.....	106

Figur oversikt

<i>Figur 1 – Organisasjonskart for Redrock AS</i>	1
<i>Figur 2 – Organisasjonskart for Redrock Tech</i>	2
<i>Figur 3 – Prosjekt livssyklus (Fritt etter Karlsen, 2013)</i>	5
<i>Figur 4 – Styringsløyfe (Fritt etter Karlsen, 2013)</i>	10
<i>Figur 5 – Fossefallsmodellen (Fritt etter Karlsen, 2013)</i>	13
<i>Figur 6 – Stegene i Scrum (Layton, 2010)</i>	17
<i>Figur 7 – Scrum Framework (nettlink 1)</i>	19
<i>Figur 8 – Prosjektnedbrytning, WBS (fritt etter Karlsen, 2013)</i>	22
<i>Figur 9 - Milepælplan med tre resultatløp (fritt etter Karlsen, 2013)</i>	23
<i>Figur 10 – CPM AON – nettverk (Link 2)</i>	24
<i>Figur 11 – AON –nettverk under PERT (fritt etter Karlsen, 2011)</i>	26
<i>Figur 12 – Gantt- Diagram (fritt etter Karlsen, 2012)</i>	27
<i>Figur 13 – Bottom – up og Top – down (fritt etter Karlsen, 2013)</i>	29
<i>Figur 14 – Cost Break Structure (Karlsen, 2012)</i>	30
<i>Figur 15 – Crashing av en aktivitet (fritt etter Karlsen, 2012)</i>	32
<i>Figur 16 – Releaseplan (Layton, 2012)</i>	36
<i>Figur 17 – Tall eksempel på et krav tidsestimerings problem (Li m.fl, 2010)</i>	39
<i>Figur 18 - S – Kurven (fritt etter Karlsen, 2013)</i>	46
<i>Figur 19 – Burndown Chart (link 4)</i>	48
<i>Figur 20 – Burndown Bar (Cohn, 2005)(Link 5)</i>	49
<i>Figur 21- Organisasjonskart for prosjektet Mobile Ticket</i>	60
<i>Figur 22 – Sykluser av arbeid i tradisjonelle - og agile metoder (fritt etter Layton, 2012)</i>	73
<i>Figur 23 – Avhengighetsforholdene mellom aktivitetene (Zsolt, 2011)</i>	76

Tabell oversikt

<i>Tabell 1 – PERT med usikkerhet (fritt etter Karlsen, 2013)</i>	26
<i>Tabell 2 – Komponenter i kostnadsestimat og budsjett</i>	31
<i>Tabell 3 – Dekomponering av produktkrav (Fritt etter Layton, 2012)</i>	34
<i>Tabell 4 – Dekomponering av produktkrav vol. 2 (Fritt etter Layton, 2012)</i>	35
<i>Tabell 5 – Avhengighetsforholdene (Li. m.fl (2010)</i>	37
<i>Tabell 6 – Eksempel på et krav- og releaseplan problem (Li m.fl, 2010)</i>	38
<i>Tabell 7 - Story points rangering (Layton, 2012)</i>	40
<i>Tabell 8 – Kvalitativ VS Kvantitativ</i>	51
<i>Tabell 9 – Sammenligning av WBS (fritt etter Karlsen, 2012)</i>	74
<i>Tabell 10 – Tradisjonell VS Agil (fritt etter Layton, 2012)</i>	77

Kapittel 1 - Innledning

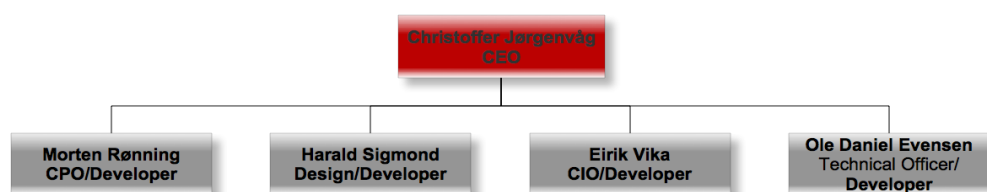
I dette kapitlet vil vi redegjøre for grunnlaget for oppgaven vår, hvor vi blant annet spesifiserer bakgrunnen for temavalget, problemstillingen og avgrensning for å sette rammer rundt oppgaven.

1.1 Mål og motivasjon for oppgaven

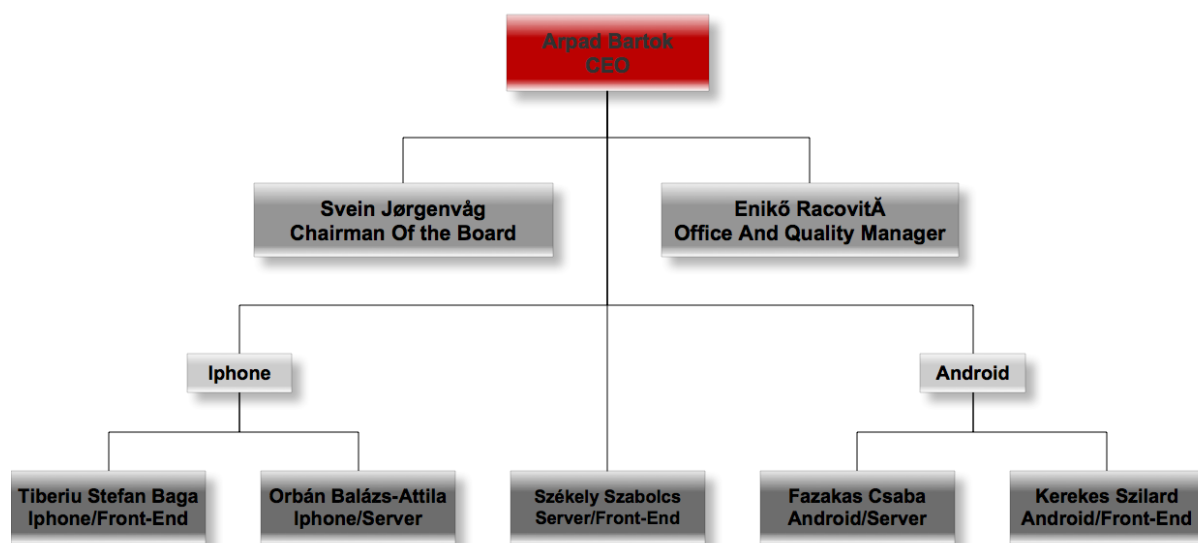
Prosjekt som arbeidsform tas ofte i bruk både i private og offentlige sektorer. Arbeidsformens store fleksibilitet, tverrfaglighet og samarbeidsmuligheter hevdes å være årsaken (Karlsen, 2013). Vår interesse for ledelse og styring av prosjekter oppstod under yrkeslivet og vi valgte derfor å fordype oss i prosjektstyring.

1.2 Red Rock Gruppen

Vi har valgt å skrive om et prosjekt som er under utvikling av konsernet Red Rock. Konsernet tilbyr produkter og tjenester innen industri, offshore, marine, IT og Telecom. Bedriften ble etablert i 2009 og består i dag av fire selskaper; Red Rock As, Red Rock TM, Red Rock DMS og Red Rock Tech. Red Rock As, heretter betegnes som RR, er et innovativt selskap for programvareutvikling innen web- og mobiltelefonapplikasjoner i tett samarbeid med Redrock Tech. Redrock Tech, heretter forkortet RRT, er et datterselskap som befinner seg i Romania. Nedenfor har vi utarbeidet organisasjonskart for RR og RRT.



Figur 1 – Organisasjonskart for Redrock AS



Figur 2 – Organisasjonskart for Redrock Tech

RR har ansvaret for å selge inn prosjekter innenfor web- og applikasjonsutvikling til potensielle kunder. De har konsentrert seg om kunder innenfor trafikkbransjen som buss og fly i forskjellige byer i Norge. Utviklingsprosjektene for kundene blir planlagt og styrt av RR og gjennomført i samarbeid med RRT. RR har ansvaret for prosjektstyringen for alle utviklingsprosjektene.

RR er et ungt selskap og har vært gjennom både store og små prosjekter. Det siste store prosjektet var å utvikle en mobilapplikasjon for busselskapet AKT i Kristiansand. Idéen var å utvikle et mobilbillettsystem hvor brukerne kunne kjøpe bussbilletter via mobiltelefonen sin. Prosjektet har vært en suksess og fått tildelt navnet Mobile Ticket som videre har blitt solgt til flere andre kunder. Flybuss Ekspress As er en av kundene som har vist interesse for produktet Mobile Ticket, og er en kunde som utgjør en stor rolle i analysen til RR. RR har inngått avtale med Flybuss Ekspress As om å videreutvikle produktet Mobile Ticket for dem. I denne oppgaven har vi valgt å analysere dette videreutviklingsprosjektet.

1.3 Problemstilling

Problemstillingen som vi har kommet frem til i masteroppgaven vår er:

Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?

Vi ønsker å analysere hvordan moderne styringsmodeller skiller seg fra de tradisjonelle. Dette bygges opp med analyse av hvordan RR styrer prosjektene sine ved bruk av moderne styringsmodeller. Hensikten med analysene er å undersøke om moderne prosjektstyringsmodeller bidrar til bedre styring. Ved hjelp av analysene ønsker vi å se om det foreligger forbedringspotensial i prosjektstyringen til RR og eventuelt hvor. Forbedringspotensialene foreslås med utgangspunkt i teorien vi har oppsummert i oppgaven vår.

1.4 Problemavgrensning

Prosjektledelse er et vidt tema. Vi beskriver her den delen av faget som skal omhandle styring av et prosjekt. Etter et par møter med RR fikk vi inntrykk av at det foreligger forbedringspotensial innenfor prosjektstyring. Vi har derfor valgt å avgrense temaet vårt til prosjektstyring med hovedfokus på sammenligning av moderne og tradisjonelle styringsmodeller. Dette gjør vi konkret ved å analysere styringen i prosjektet Mobile Ticket til RR.

Kapittel 2 - Teori

I dette kapitlet vil vi redegjøre for teorier innenfor prosjektstyring som vi mener vil være aktuelle ved analyse av prosjektet Mobile Ticket. Vi starter med å definere prosjekt og andre viktige prosjektbegreper. Deretter ser vi på prosjektstyringen i både tradisjonelle - og agile metoder for å se hvordan metodene skiller seg fra hverandre.

2.1 Prosjekt

Begrepet prosjekt kommer fra det latinske ordet ”proicere” som betyr å kaste frem. Det representerer en plan, et forslag eller et utkast og er temporært (Karlsen, 2013).

Ifølge PMI – retningslinjene (PMI, 2008) handler prosjektledelse om:

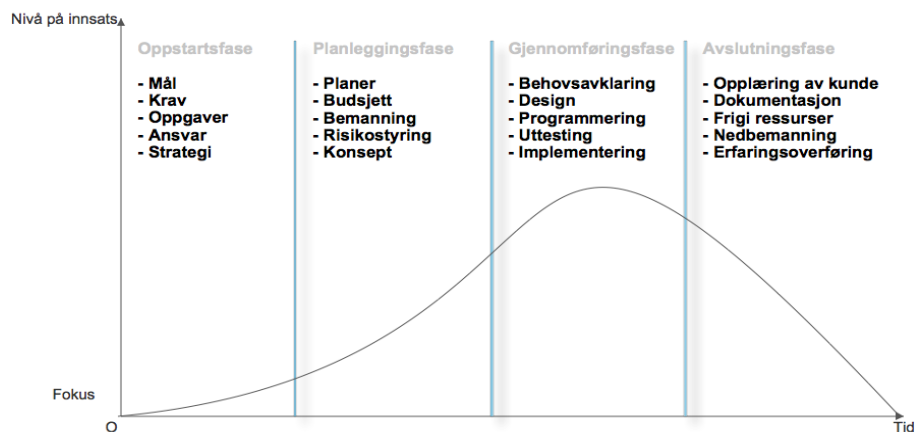
Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.

Det er fem karakteristiske trekk ved et prosjekt (Karlsen, 2013):

- Entydig målsetting – unik målsetting som forteller hva som skal oppnås og hvorfor.
- Begrenset ressurstilgang – definert som et pengebeløp
- Engangsforetak – prosjektet sett på som helhet er unik
- Bestemt start og sluttdato – når arbeidet skal starte og avsluttes
- Tverrfaglig arbeid – involvering av personer med både forskjellige kunnskap, erfaring og administrativ tilhørighet.

Prosjektlivssyklus

Modellen i figur 3 viser hvordan prosjekter kan ledes gjennom dets ulike faser.



Figur 3 – Prosjekt livssyklus (Fritt etter Karlsen, 2013)

Oppstartsfasen

I oppstartsfasen setter man mål og visjon for prosjektet og produktet som skal utvikles. Kunden kommer med krav om hva de ønsker at produktet skal bidra med. Prosjektgruppen må da analysere og sette opp oppgaver som må gjennomføres for å nå dette målet. En strategi for gjennomføringen må legges klar. Deretter delegeres ansvaret av oppgavene til hver enkelt.

Planleggingsfasen

I denne fasen utarbeides det et budsjett og planlegging av oppgavegjennomføringen som må foretas. Det går ut på å kartlegge hvilke ressurser som trengs til oppgavene, og klargjøre bemanningen som trengs. Ved alle prosjekter følges det en viss risiko som bør analyseres og legges en strategi for hvordan bedrifter skal styre prosjektet og håndtere uforventede hendelser som kan oppstå.

Gjennomføringsfase

I denne fasen arbeides det med å få prosjektet gjennomført. Her settes det opp løpende statusrapporter for hvor langt i prosjektet bedriften har kommet. Dersom det kreves endringer må det rådføres med prosjektleder, eventuelt produkteier eller kunde og sørge for at prosjektet kvalitetssikres. Framdriftsrapporter må avlegges for å se hvordan bedriften ligger i forhold til planen og eventuelt avdekke avvik. Framdriftsrapporten skal både fungere som en drivkraft- og oppfølgingsverktøy.

Avslutningsfase

I avslutningsfasen er produktet utviklet og klar for levering til kunden. Dokumentasjon eller veiledning til produktet må lages. Her frigis ressursene og prosjektet nedbemannes. Erfaringen som har kommet av prosjektet bør overføres til brukerne og eventuelt dokumenteres for senere opplæring.

2.1.1 Prosjektmandat

Prosjektmandat er et dokument som oppsummerer defineringsene av prosjektet og bør inneholde følgende punkter (Karlsen, 2013):

- **Prosjektets navn og bakgrunn** – et navn som kan identifiseres med prosjektet. Bakgrunn kan i dette tilfellet være behovet som prosjektet bygger på.
- **Oppdragsgiver** (prosjekteier) – er den som er ansvarlig for prosjektet, som eksempelvis kan være linjesjefen eller en ekstern kunde.
- **Prosjektets for-, effekt- og resultatmål** – disse målene skal til sammen beskrive hva resultatet skal bestå av. *Formål* skal gi oss et bilde av hva hensikten med prosjektet er og hvorfor. *Resultatmål* sier noe om hva prosjektet konkret skal få til i løpet av prosjektperioden, og *effekt* er et mål man setter for bruk av prosjektets resultater.
- **Arbeidsomfang og krav til løsninger** – hva prosjektet omfatter og hva som ikke inngår.
- **Rammebetingelser** – er betingelser som prosjektledelsen må være oppmerksom på og ta hensyn til. Det kan bestå av budsjett- og tidsrammer.

Det kan også være noen interne restriksjoner (som for eksempel ressursbruk, uakseptable løsninger og tiltak) og eksterne restriksjoner (som for eksempel fysisk plassering, forskrifter og lover).

- **Roller og ansvar** – definere og klargjøre roller - og ansvar beskrivelser for alle som er involvert i prosjektet.

Suksessfaktorer og -kriterier

Suksessfaktorer er noe prosjektlederen sammen med medarbeiderne må tilføre prosjektet for å nå suksess. Dette kan for eksempel være god planlegging, god oppfølging, god prosjektledelse, god usikkerhetshåndtering osv. Her er det viktig å få diskutert og vurdert de mest relevante og viktigste faktorene for prosjektet, da dette kan bli en lang liste. Faktorene vil ofte avhenge av prosjekttype. Derfor bør man ha god kjennskap til prosjektets mandat før man vurderer disse.

Suksesskriterier er kriterier som brukes i etterkant av prosjektets gjennomførelse til å vurdere hvor vellykket prosjektet har vært. Man tar utgangspunkt i prosjektets produktkrav, som er utformet av dets interessenter, for å finne suksesskriterier for et prosjekt. Det er interessentene som avgjør hvor vellykket prosjektet har vært. De tre suksesskriteriene som har vært mest dominerende i prosjektlitteraturen er KKT; kostnad, kvalitet og tid.

2.2 Prosjektstyring

Styring kan defineres som teknikker for å planlegge og følge opp prosjektarbeidet (Rolstadås, 2011). Prosjektstyring er en av hovedoppgavene ved ledelse av et prosjekt, og er et grunnleggende verktøy for å håndtere hvordan et prosjekt skal gjennomføres i forhold til tid, penger og andre ressurser. Dette innebærer planlegging, organisering og ressursstyring som er nødvendig for å oppnå definerte prosjektmål. Prosessene i styringen innebærer blant annet å foreta nødvendige beslutninger og vurdere ulike tiltak for å sikre at aktivitetene med god kvalitet blir fullført i henhold til planen. Med andre ord kan styringen betraktes som en kontinuerlig prosess.

Styringsoppgavene kan deles inn i to grupper (Rolstadås, 2011):

- 1) Planlegging
 - formulere mål
 - utarbeide fremdriftsplan
 - beordre plan

- 2) Oppfølging
 - registrere fremdrift
 - iverksette korrektive tiltak

Planlegging går ut på å utarbeide en plan for hvordan prosjektet skal gjennomføres.

Det innebærer å planlegge arbeidsomfang, tid og kostnader på flere nivåer.

Oppfølging går ut på å måle og sammenligne planen for å kontrollere at alt går som planlagt og registrere eventuelle avvik. Dersom det oppstår avvik bør ulike tiltak vurderes for å sikre at målene oppnås (Rolstadås, 2011).

2.2.1 Styringsnivåer og former

Styringen kan deles inn i to nivåer (Karlsen, 2013):

- Strategisk styring – handler om å definere prosjektets mål, kartlegge milepæler, bestemme kontraktsstrategi, og hvordan håndtere forholdet til interessentene.

- Operativ styring – handler om å styre den daglige driften av prosjektet og utnytte de ressursene som er stilt disposisjon for prosjektet på en best mulig måte. Det kan være i form av analysering av framdriftsrapporter.

I følge Karlsen (2013) kan styringen av et prosjekt skje på forskjellige måter, og alternativene er som følger:

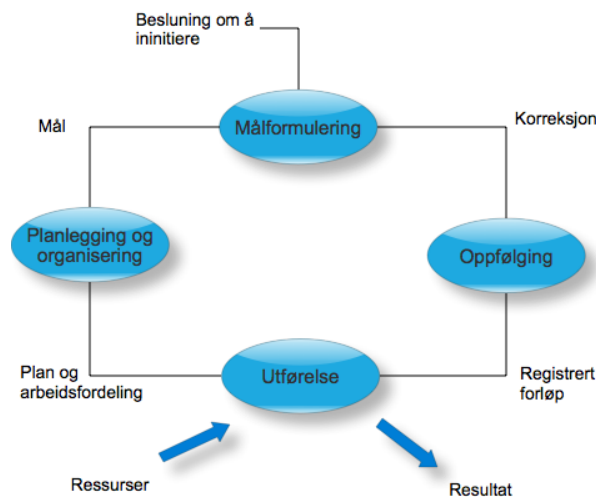
- *Direkte styring* går ut på at den overordnede leder gir direkte instruksjoner eller ordrer til den underordnede om hva og hvordan en oppgave skal gjøres. *Fordelen* er at det er tydelig og klart og liten mulighet for

misforståelse. *Ulempen* er at det gir liten handlefrihet for den enkelte utøver.

- *Regelstyring* innebærer at det eksisterer et sett av regler og instruksjoner som gir detaljerte føringer for hvordan arbeidet skal utføres. Dette kan være regler eller prosedyrer for rapportering og oppfølging. *Fordelen* ved det er at man får en standardisering av hva og hvordan det skal gjøres. Muligheten for misforståelsen er lav. *Ulempen* kan være liten handlefrihet for den enkelte utøveren.
- *Målstyring* knytter styringen til de mål og resultater som ønskes oppnådd, for eksempel spesifisert mandat og budsjetter. *Fordelen* er at det gir vesentlig grad av handlingsfrihet og medvirkning i beslutninger innenfor de gitte rammebetingelsene. *Ulempen* ved denne formen kan være at målet i seg selv kan være oppslukende.
- *Kulturstyring* er en styringsform som skjer ved hjelp av verdier og organisasjonskultur. Her styres det gjennom å etablere og vedlikeholde riktig kultur. En *positiv side* er at en organisasjon med godt utviklet prosjektkultur vil frembringe gode resultater. På *minussiden* vil det trekke frem at det kan være vanskelig eller ta lang tid å få etablert den ønskede og riktige kulturen i organisasjonen.

2.2.2 Styringsfunksjoner og -faktorer

Styringsfunksjoner og -faktorer kan illustreres i en modell, som kalles for *styringsløyfe*. Figur 4 viser prosjektets styringsløyfe, fra målformulering til ferdigstillingen (Karlsen, 2013). Sløyfen starter med *målformulering* som går ut på å beslutte og beskrive hva prosjektet skal resultere i. *Planlegging og organisering* innebærer å bryte målene ned i arbeidsoppgaver og deretter planlegge og organisere utførelsen av arbeidsoppgavene. I *utførelsesfunksjonen* skjer selve gjennomføringen av arbeidsoppgavene. *Oppfølging* er den siste funksjonen som går ut på å registrere, måle, og eventuelt korrigere ved hjelp av tiltak og deretter evaluere prosjektgjennomføringen.



Figur 4 – Styringsløyfe (Fritt etter Karlsen, 2013)

Styringsløyfe benyttes på alle prosjektets styringsfaktorer:

- Kvalitet, egenskaper og ytelse (kvalitetsstyring)
- Tid (fremdriftsstyring)
- Økonomi (økonomistyring) og ressurser (ressursstyring)
- Arbeidsomfang (endringsstyring)

Dagens grunnlag for prosjektstyring bygger på tre grunnpilarer (Rolstadås, 2011):

- Prosjektnedbrytning (WBS)
- Nettverksanalyse (PERT eller CPM) (tidsplanlegging)
- Fremdriftsmåling ved hjelp av inntjent verdi (oppfølging)

Disse vil bli redegjort for i de neste kapitlene.

2.2.3 Fremdriftsplan

Fremdriftsplanen er grunnlaget for effektiv styring og er et hjelpemiddel som synliggjør hvordan man kan styre et prosjekt for å oppnå prosjektets mål, sett i et tidsperspektiv (Karlsen, 2012). Overordnet fremdriftsplanlegging innebærer å få oversikt over ressursene for allokering, deretter fordele arbeidet og følge progresjonen. Planlegging på detaljnivå innebærer å bestemme varigheten av hver enkelt aktivitet, og hvordan disse aktivitetene skal plasseres i forhold til hverandre i tid. Dette innebærer å få oversikt over og sette av ressurser, fordele arbeidet og følge

opp fremdriften. En fremdriftsplan bør både være realistisk og ambisiøs for å kunne gi en ønsket fremdrift for utviklingsprosessen fram til implementeringen. Dersom det ikke er nok ressurser eller nok motivasjon vil det føre til at fremdriften hindres.

2.2.4 Endringshåndtering

Å ha kontroll over endringene i et prosjekt er en forutsetning for suksess i prosjektet. Dette fordi endringer forbindes med konsekvenser som blant annet kan gi økning i kostnader, forsinkelser, redusert produktivitet osv (Rolstadås, 2011).

Rolstadås (2011) skiller mellom to typer endringer:

- *De som skyldes en utvidelse av prosjektet* medfører at forutsetningene eller rammebetingelsene for prosjektet blir endret. Det kan for eksempel skyldes nye krav fra produkteieren, offentlig pålegg eller vesentlig forskjellige betingelser for gjennomføring enn forutsatt.
- *De som faller innenfor opprinnelig prosjekt* er endringer som man normalt opplever i alle prosjekter ved at nye detaljer avdekkes etter hvert som prosjektet skrives frem. Enkelte av disse endringene kan være i konflikt med, eller ha vanskeligheter med å innpasse, i det som tidligere har blitt gjort.

IT-prosjekter er kjent for å ha omfattende designendringer. For å holde oversikten over slike endringer har det blitt utviklet et eget fagområde, kalt for *konfigurasjonsstyring* (Rolstadås, 2011). Konfigurasjonsstyring har fokus på å identifisere og følge opp funksjonelle og fysiske designkarakteristikker til et system. Konfigurasjonsstyring består av fire trinn (Rolstadås, 2011):

1. Identifikasjon – identifisering av konfigurasjonsenheter.
2. Endringsstyring – evaluere endringen.
3. Statusregnskap – registrering og oppfølging av endringene.
4. Revisjon – demonstrasjon av det ferdige produktet med endringene som er foretatt.

2.3 Styringsmodeller

Det finnes hovedsakelig to metoder for å styre prosjekter på; tradisjonelle og agile metoder.

2.3.1 Tradisjonelle metoder

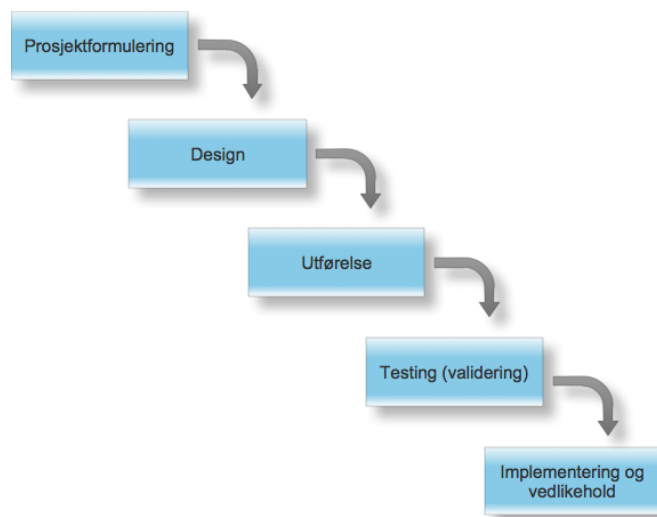
Tradisjonelle prosjektledelsesmodeller kan defineres som lineære eller en inkrementelle tilnærming til prosjektgjennomføring som består av fem primære faser eller prosessgrupper: prosjektdefinering, planlegging, lansering, overvåking og kontroll, og avslutt prosjektet. (Wysocki, 2009). Disse aktivitetene er sekvensielle, og blir altså utført i en bestemt rekkefølge.

Fossefallsmetoden

Fossefallsmetoden er en av de mest kjente metodene innenfor tradisjonelle prosjektmodeller. Denne metoden har sin opprinnelse fra bygge- og anleggsindustrien, og beskriver en lineær og sekvensiell utvikling. Fasene i fossefallsmetoden har tydelige mål og beskrivelser for hvordan aktivitetene skal utføres. Når en fase er fullført, fortsettes det videre til neste fase. Da sies det å ikke ha en vei tilbake. Hvordan fasene blir inndelt varierer fra prosjekt til prosjekt og fra organisasjon til organisasjon. Noen prosjekter har kun behov for to faser, som konseptutvikling og gjennomføringsfase. Andre prosjekter krever en finere og mer definert inndeling.

Fossefallsmodell

Figur 5 viser et eksempel på hvordan et IT-prosjekt kan deles inn i faser med utgangspunkt i fossefallsmetoden (Thakurta, 2012).



Figur 5 – Fossefallsmodellen (Fritt etter Karlsen, 2013)

Figur 5 viser fem klare faser i et prosjekt som fortsetter sekvensielt. I denne modellen er prosessene og resultatene definerte, noe som gjør at det blir lettere å lede et prosjekt som baserer seg på denne modellen. Modellen egner seg derfor best for prosjekter som har veldefinerte krav og spesifikasjoner.

I denne metoden investeres det mye tid med kunden i de første fasene som for eksempelvis i prosjektformulerings- og design fasene. Dette er med på å redusere risikoen for endringer i prosjektet, da det hevdes at metoden ikke har en løsning for å gå tilbake til de tidligere prosessene. Under systemutvikling og implementering er det lite kundeinvolvering, da disse fasene gjennomføres i henhold til de tidligere fasene. Dette øker risikoen for at løsningen ikke møter kundenes forventninger, som kan ha endret seg med tiden.

Tradisjonelle metoder er sekvensielle med fastlagte planer for hver fase. Denne metoden hevder å ikke egne seg for IT prosjekter, da det forekommer store endringer under utviklingsprosessen (Baird and Riggins, 2012).

2.3.2 Agile metoder

Begrepet Agile kommer fra av det engelske ordet "Agility" som på godt norsk betyr smidighet. Det å være smidig kan betegnes med å gjøre en rask handling og å kunne endre kurs når det trengs. Agile metoder ble formet som en reaksjon tradisjonelle metoder da de ble sett på som veldig omfattende og tregt.

Bakgrunnen for agile metoder bygger av den iterative- og inkrementelle utviklingen(Larman, 2004). En iterasjon har gjerne en relativ kort tidsramme hvor det skal utføres en rekke aktiviteter, eksempelvis analyse av produktkrav, design, implementering og testing. Avslutningen av en iterasjon skal bli en del av et prosjekt. Når man bygger på disse iterasjonene med nye produktkrav og/eller design vil det bli en inkrementell utvikling, som kommer av at ting blir bygd opp på hverandre litt etter litt. Den iterative- og inkrementelle utviklingen er blitt sporet tilbake til 1960 som en reaksjon på den tradisjonelle metoden for programutvikling. Om man går enda lengre tilbake som til 1950-årene kan man finne røttene til den iterative- og inkrementelle utviklingen som et alternativ til den voksende fossefallsmodellen(Larman, 2004)

Ifølge Layton (2012) kan agil prosjektledelse defineres som:

Agile project management is a style of project management that focuses on early delivery of business value, continuous improvement of the project's product and processes, scope flexibility, team input, and delivering well-tested products that reflect customer needs.

Det agile manifestet

Personer og samspill fremfor prosesser og verktøy
Programvare som virker fremfor omfattende dokumentasjon
Samarbeid med kunden fremfor kontraktsforhandlinger

Den uthevede skriften markerer at selv om punktene som står til høyre har verdi, så verdsetter vi punktene til venstre enda høyere (Beck m.fl., 2001).

Verdiene i manifestet bidrar til å heve kvaliteten og redusere risikoen for stor kompleksitet i et prosjekt, deretter for å øke engasjementet gjennom god endringshåndtering. Enkelhet er sett på som essensielt i agil prosjektarbeid.

Idéene fra ”Det agile manifestet” kan implementeres for design-, data-, og noen forskningsprosjekter, men noen hevder at i fysiske tilvirkningsprosjekter kan dette grunnlaget bli for enkel. Modellen kan kanskje kalles en designmetode enn en komplett prosjektmodell, da den ikke er egnet i gjennomføringsfasen for komplekse tilvirkningsprosjekter.

Scrum

Scrum er en av styringsmodellene innenfor agile metoder og er den mest anvendte metoden blant de agile programutviklingsmetodene. Denne metoden er spesielt tilpasset prosjekter der man ikke kan forutse, forstå eller definere utfordringer i et prosjekt fullt ut i forkant. Scrum fokuserer på å utvikle evnen til teamet å gi raske levering, være i stand til å håndtere endringer i forutsetninger, og kunne justere kursen underveis i prosjektet. Derfor kjører man iterasjoner i Scrum hvor fokuset er å kunne levere løsninger i henhold til krav og spesifikasjoner og deretter blir dette demonstrert for kunden (Cohn, 2010).

Hovedroller

Produkteieren kan være betegnelsen for enten kunderepresentanten, organisasjonen som utvikler prosjektet, eller den som representerer alle interessentene til prosjektet. Hovedoppgave til produkteier er å utarbeide og presentere produktets overordnede krav, og sette det inn i produkt backloggen med høyst prioritert produktkrav øverst som skal utvikles først av *Scrum teamet*. Dersom det oppstår endringer av produktkravene underveis i utviklingen av produktet er det produkteierens oppgave å oppdatere produktbackloggen.

Scrummaster fungerer som en portvakt. Hovedoppgavene til scrummasteren er å tilføre teamet bistand til problemløsninger slik at de får arbeidet mest mulig uforstyrret og kommer i mål. Rollen som scrummaster har i seg selv ikke autoriteten som rollen til en prosjektleder. Han er en del av teamet selv om han kanskje ikke er en av utviklerne.

Scrumteamet består av utvikler som har hovedansvaret for å utvikle selve produktet. Teamet består gjerne av 5 – 9 personer med ulike kompetanser. Teamet er selvorganiserende, og er selv ansvarlige for å planlegge og gjennomføre oppgavene i henhold til planen. Medlemmene besitter selv med myndigheten til å ta de beslutninger som behøves for å nå målet.

Begrepsdefinisjoner

I Scrummetodikken finnes det ulike verktøy og begreper. De som er sentrale for å forstå scrummodellen er som følgende:

Produkt backlogg; består av produktkravene for et endelig produkt som er ønskelig av produkteieren, i en prioritert rekkefølge. Produkteieren skal sammen med Scrum teamet stå ansvarlig for å oppdatere og sørge for at den blir fulgt opp. Backloggen brukes som et utgangspunkt i planleggingen av sprinter. De mest prioriterte produktkrav blir valgt ut først og innarbeidet i sprinter.

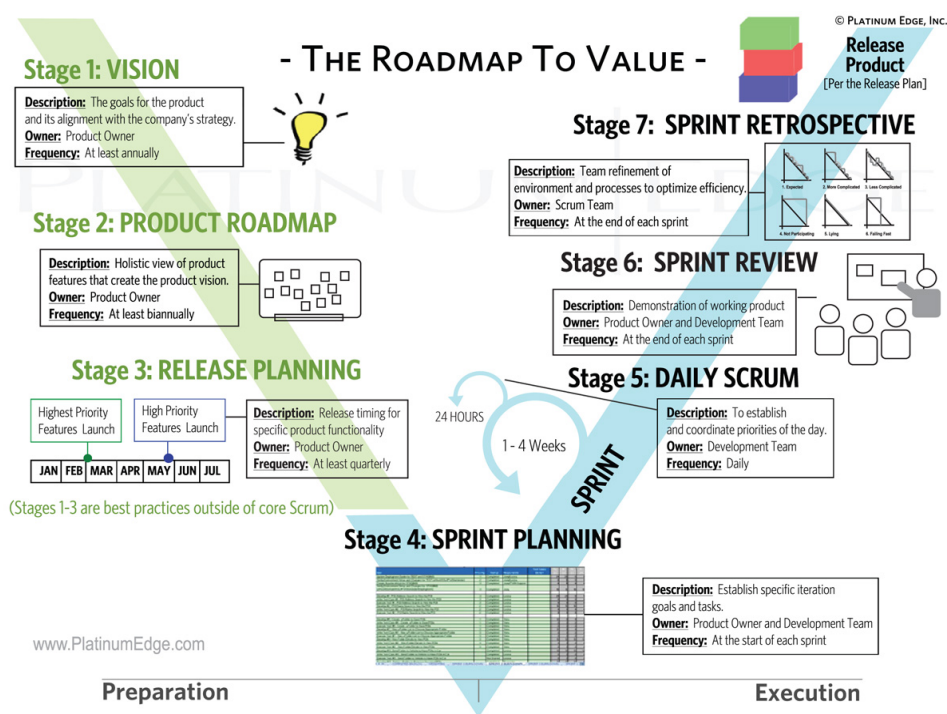
Sprint er en iterasjon med en optimal lengde på mellom 1 - 4 uker, men den kan tilpasses ut i fra hvilken fase prosjektet er i. Dette er perioden hvor teamet kan prøve og feile uten at det føres dokumentasjon. Interessentene av prosjektet anbefales å få løpende demonstrasjoner av utviklingen mellom sprintene slik at de kan komme med tilbakemeldinger underveis. Under sprinten arbeider teamet fritt og kan rådføre seg med for eksempel produkteieren eller andre utviklere for å få en oppgave løst.

Sprint backlogg er en iterasjonsplan som inneholder oppgaver som teamet skal fullføre i løpet av en sprint. Disse oppgavene blir utarbeidet ut fra produktkravene

som blir valgt ut fra produktbackloggen. De mest kritiske oppgaver blir håndtert først. Dette kan for eksempel være oppgaver som tar mye tid å utføre på grunn av kompleksitet eller mangel på informasjon som må hentes eksternt. Det kan også være oppgaver som produkteier eller kunde krever å få en demonstrasjon av i første omgang. Sprintbackloggen oppdateres underveis til informasjon for interessenter og produkteier.

Roadmap

Figur 6 viser en oversikt over alle faser i scrummetoden. Fasene i figuren er delt i to hoveddeler; Forberedelsesdelen og gjennomføringsdelen. Faser fra 1- 3 er felles for alle agile metoder, mens faser fra 4-7 er spesifikt for scrummetoden.



Figur 6 – Stegene i Scrum (Layton, 2010)

- 1) Forberedelsesdelen** består av tre faser som skjer utenfor Scrum kjernen og som vanligvis blir utført av produkteieren/ kunderepresentanten sammen med utviklingsteamet.

I fase 1 blir produktets visjon identifisert av produkteieren, altså hva produktet er og hvordan den skal støtte bedriftens strategi og ikke minst hvem som skal bruke den. I fase 2 utarbeides det en slags veikart ”product roadmap” av

produkteieren og utviklingsteamet. Veikart gir en oversikt over produktets krav, funksjoner og spesifikasjoner med en løs tidsramme.

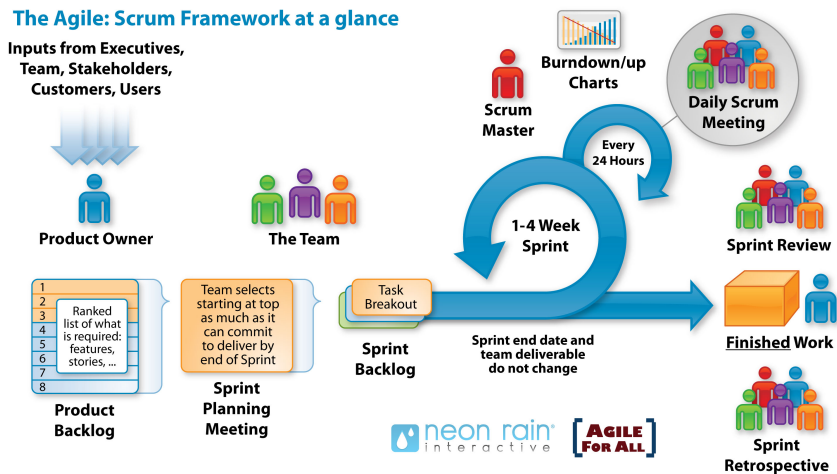
I fase 3 blir releaseplaner, utgivelsesplaner, utarbeidet. En releaseplan er en tidsplan for utgivelsen av delprodukter som for eksempel kan være en fungerende del av en programvare. Agile prosjekter kan inneholde flere utgivelsesplaner og hver utgivelsesplan kan inneholde fra tre til fem sprints.

- 2) Utviklingsdelen** består av 4 faser som gjenspeiler scrummetodens livssyklus (se figur 15). Disse fasene blir repetert for alle sprintene i en releaseplan.

Fase 4 går ut på å planlegge sprintene på detaljplan ut fra den utvalgte releaseplanen. Dette starter med å bestemme sprintvarighetene, deretter starter utviklingsteamet med utviklingen i henhold til sprintplanen. Sprint planleggingen finner sted ved starten av hver sprint. I løpet av hver sprint har utviklingsteamet daglige Scrummøter på ca. 15 min som er *fase 5*. I møtene blir blant annet gårsdagens arbeid oppsummert og dagsoppgavene plannlagt, om hva som skal bli gjort osv. På slutten av hver sprint blir det holdt ”sprint review” møter som skjer som er *fase 6*. I disse møtene blir delproduktet, som er laget under sprinten, demonstrert til prosjektets interessenter dersom det er ønskelig. *I fase 7* drøftes det om hvordan utviklingen har gått i sprintene har og planlegger eventuell forbedringer til neste sprint. Denne fasen blir kalt for sprintens retrospektiv/evaluering. Dette holdes på slutten av hver sprint før det begynnes på en ny sprint.

Fase 3 og 4 kan ses på som en fremdriftsplan og vil bli redegjort i detalj i under delkapitlet; planlegging av Agile metoder.

Figur 7 illustrerer utviklingsfasene som gir en totaloversikt over Scrumroller og dets innspill i ulike stadiene.



Figur 7 – Scrum Framework (nettlink 1)

Vanlige feil og misforståelser ved Scrum i styring og gjennomføring

Scrum-metodikken som RR benytter er en av de eldste agile metodene som eksisterer. I lys av dette har det blitt forsket på hvorfor Scrum feiler som styring-/gjennomføringsmetode. Følgende punkter er feil som begås og fører til at Scrum feiler (Larman, 2004):

- Det er ikke et selvstyrende team; prosjektleder eller Scrum-master styrer eller organiserer teamet – mange prosjektledere er vant til å planlegge og lede, og har derfor vanskeligheter med å motstå trangen til å ta over ledelsen av teamet. Mange har derfor problemer å tilpasse seg Scrum-master rollen med å fjerne hindringer raskt, skaffe ressurser, fungere som en brannmur mot resten av organisasjonen, men ellers la teamet styre selv. Spesielt gjelder dette i Scrum-møter, hvor det er en naturlig sammenheng å se til lederen for styring.
- Ingen daglige oppdateringer av sprintbacklogg av medlemmene eller daglig sporing – med andre ord ingen statusoppdateringer, vanskelig da å følge fremdriften.
- Nytt arbeid lagt til iterasjonen eller utvikler – selv i et hav av konstante endringer må det være form for stabilitet. Når en sprint har begynt, bør det ikke legges til nye produktkrav, er Scrums måte å holde kontroll på.

- Produkteier er ikke involvert eller ikke bestemmer – Scrum er drevet av kunden; produkteieren må være med å bestemme hva som skal prioriteres i produktbackloggen og velge produktkravene som skal utføres i neste sprint
- Ikke noe sprint evaluering – Tilbakemelding og tilpasning er det som driver Scrum; demoen og evalueringen er nødvendig for informere kundene slik at de kan planlegge neste iterasjon
- Mange ledere: Det er kun produkteieren som skal styre produktkravene i produktbackloggen, prioriteringene og arbeidet i neste sprint
- Dårlig dokumentasjon – Scrum er ikke imot dokumentasjon, diskusjon av prosjekt arbeidsprodukter er bare utenfor deres område av definisjon. Som med alle agile metoder, er ikke-kode arbeidsprodukter forventet å gi en reell verdi, enn å bli laget for bare å følge en prosess formel.
- Hele teamet (inkludert kunder og ledelse) er ikke undervist i Scrum og dets verdier – det sier seg selv at man kan da ikke jobbe med Scrum
- For lange Scrum-møter eller for ufokuserte – møtene skal være under 20 minutter, helst mellom 10-15 minutter, og det må være fokus på scrum-spørsmål
- Forutseende planlegging; Planlegging av PERTH diagram: Som med alle iterative og inkrementelle metoder er det misforståelse å lage en plan som legger ut nøyaktig hvor mange sprints det skal være for et langsiktig prosjekt og hva som vil fremkomme i disse, eller å lage PERT diagram som identifiserer mange oppgaver, deres rekkefølge og estimere varighet.

2.4 Planlegging i tradisjonelle modeller

Tradisjonelle metoder beskriver bestemte og sekvensielle fremgangsmåter for hvordan prosjekter skal planlegges og oppfølges. I denne metoden benyttes det standardisering og målemetoder ved hjelp av konkrete verktøy, teknikker og praksiser.

2.4.1 Prosjektnedbrytning (WBS)

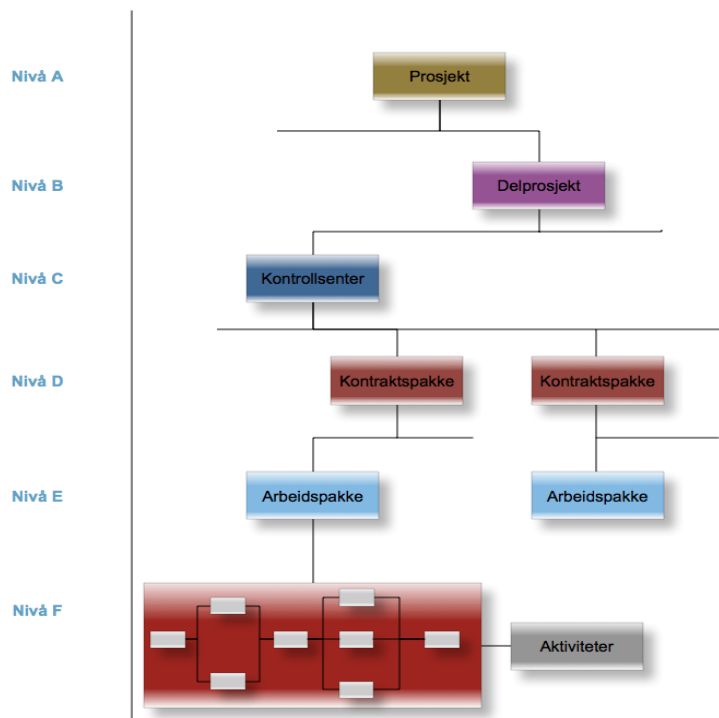
For å kunne planlegge ressurser og varighet for et prosjekt er man først nødt til å bryte opp hele prosjektet i arbeidsoppgaver.

Prosjektnedbrytning, også kjent som Work Breakdown Structure (WBS), er en fundamental forutsetning for prosjektstyring og innebærer at arbeidsoppgaver blir oppdelt i klart avgrensede og håndterbare arbeidsenheter (Rolstadås 2011). Denne oppdelingen gir oss bedre forståelse av hva prosjektet består av og dermed synliggjøre hva som skal gjøres av arbeid i prosjektet. Arbeidsomfanget i prosjektet vil derfor kunne defineres gjennom WBS og tilhørende ressursbehov for hver enkel oppgave. Prosjektnedbrytning er nødvendig for en effektiv oppfølging av prosjektgjennomføringen.

Nedbrytningen er med på å angi to viktige forhold (Karlsen, 2012):

- Hvilke oppgaver som inngår i prosjektet
- Hvilke relasjoner som gjelder mellom disse oppgavene (struktur)

Figur 8 viser typiske nivåene i en WBS. Milepælene befinner seg i arbeidspakke nivået som deretter deles opp i aktiviteter:



Figur 8 – Prosjektnedbrytning, WBS (fritt etter Karlsen, 2013)

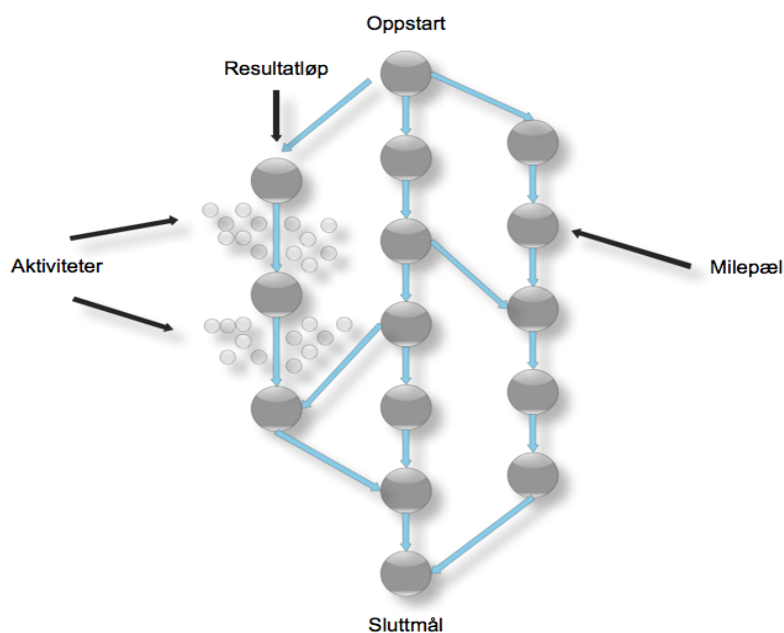
2.4.2 Milepælmetoden – overordnet nivå

Milepælplanlegging er en metode for tidsplanlegging på overordnet arbeidspakkenivå som gir en oversikt over prosjektets retning (hva, når og hvilke ressurser som skal brukes). Ifølge Karlsen (2012) kan en milepælplan defineres som:

En milepælplan er en overordnet plan av strategisk karakter som setter retning for prosjektet. Den utgjør et rammeverk og består av mindre resultater og leveranser som skal nås i løpet av prosjektperioden. Planen viser en logisk sekvens av tilstander som prosjektet skal igjennom på veien mot å nå sluttmålet, og det kan ofte være knyttet betingelser til hvordan milepælene skal oppnås (Karlsen, 2013).

Milepælplan kan beskrives som en tilstand som prosjektet bør være i på et visst stadium av prosjektarbeidet. Milepælene kjennetegnes ved at de viser hva som skal oppnås og ikke nødvendigvis hvordan, og kan derfor ses på som en kontrollstasjon.

I figur 9 er det illustrert et eksempel på milepælplan med tre resultatløp med start og slutt milepæl, og milepælindelning i aktiviteter. Parallele gjennomføringer av prosjektets aktiviteter kan bidra til å redusere prosjekttiden. Koblingene mellom resultatløpene indikerer at de forskjellige resultatene er avhengige av hverandre. Det anbefales å ikke ha mer enn 3 måneders mellomrom mellom milepælene og helst ikke være over 25 milepæler totalt i prosjektet (Karlsen, 2012).



Figur 9 - Milepælplan med tre resultatløp (fritt etter Karlsen, 2013)

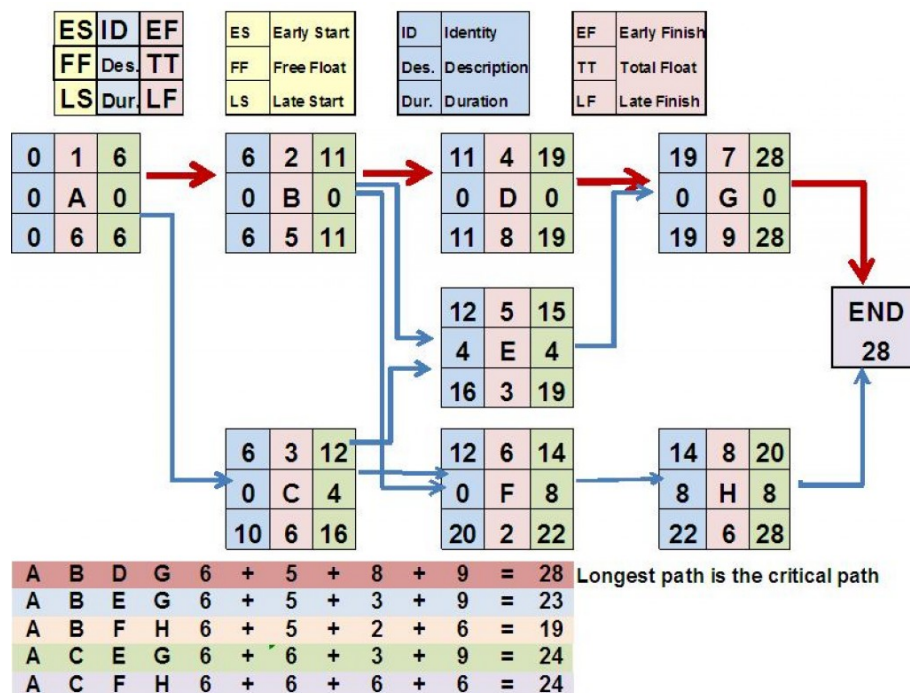
2.4.3 Critical Path Method (CPM)

CPM er en detaljert planleggingsmetode på aktivitetsnivå (Karlsen, 2013).

Aktiviteteene identifiseres ved å bryte opp hver arbeidspakke i flere aktiviteter og tidsplanlegger det ved hjelp av et nettverk. Nettverket kan fremstilles på to forskjellige måter AON – activity on node og AOA – activity on arch. Vi har valgt å fokusere på AON, da vi oppfatter dette nettverket som mer oversiktlig.

Figur 10 viser et eksempel på et AON nettverk. Modellen i figuren viser en oversikt over en arbeidspakke som inneholder aktiviteter fra A til H og har identitet fra 1- 8. En aktivitet representerer en samling av arbeidsoppgaver som krever tid og ressurser

for å bli utført. Helt øverst i modellen er det en oversikt over hva de ulike forkortelsene og tallene beskriver.



Figur 10 – CPM AON – nettverk (Link 2)

Vi ser at A (6 tidsenheters varighet) er en startaktivitet og B (5 tidsenheters varighet) og C (6 tidsenheters varighet) er aktiviteter som kan starte umiddelbart etter A. Derfor får begge aktivitetene tidligst start etter 6 tidsenheter. Aktivitet D (8 tidsenheters varighet) kan starte umiddelbart etter at B er ferdig og får dermed en tidligst start etter 11 tidsenheter. Aktivitetene E (3 tidsenheters varighet) og F (2 tidsenheters varighet) kan derimot starte når både B og C er ferdig. Da aktivitet C blir ferdig etter B får aktivitetene E og F tidligst start på 12 tidsenheter. Aktivitet G (9 tidsenheters varighet) kan starte når D og E er ferdig. Selv om E er ferdig før D må G vente på D og får dermed tidligst start etter 19 tidsenheters varighet. Både G og H er betegnet som sluttaktivitet, men begge må være ferdige for en endelig slutt på arbeidspakken. Aktivitet G blir ferdig etter 28 tidsenheters varighet, mens aktivitet H blir ferdig etter 20 tidsenheters varighet. Det vil derfor ta totalt 28 tidsenheters varighet for å gjennomføre alle aktivitetene (END).

Nederst på figur 10 er det vist en oppsummering av alle stiene med tilhørende varighet gjennom nettverket. Vi ser at den første stien, A – B – D – G = 28, er den

lengste stien og betegnes som kritisk sti. En *kritisk sti* er den stien som tar lengst tid gjennom nettverket. Blir en eller flere av aktivitetene på kritisk sti forsinket, blir prosjektet som helhet forsinket med samme tid (kalles også for kritisk vei eller kritisk linje). Denne kritiske stien kan vi også finne ut ved å gå motsatt vei i nettverket. Vi kan deretter beregne flyten i aktiviteter som ikke er på kritisk sti ved å beregne følgende regnestykke: tidligst ferdig (LF) ÷ tidligst ferdig (EF) = Flyt. *Flyt (slakk)* forteller oss hvor mye en gruppe aktiviteter på en ikke-kritisk linje til sammen kan utsettes i tid uten at det påvirker prosjektets varighet. Aktiviteter med 0 i flyt er tegn på at de befinner seg på den kritiske stien.

2.4.4 Programme evaluation and review technique (PERT)

PERT er en ny styringsmetode som bygger på CPM. Denne tidsplanleggingsmetoden er nesten identisk med CPM, men skiller seg ved at den tar hensyn til usikkerhet. I denne metoden antar man at en aktivitets varighet har en variasjonsbredde som følger en statistisk distribusjon. Dette går ut på at man estimerer tre tidsestimater for hver aktivitet (Karlsen, 2013):

- *Optimistisk tid (a)* – varigheten til en aktivitet når forholdene er ideelle
- *Mest sannsynlig tid (m)* – varigheten til en aktivitet ved normale forhold
- *Pessimistisk tid (b)* – varigheten til en aktivitet når en tar hensyn til at flere forhold som kan gå galt, virkelig går galt

Man bruker deretter de tre overnevnte estimatene til å beregne forventet varighet og variansen for aktivitetene ved hjelp av følgende formler (Karlsen, 2013):

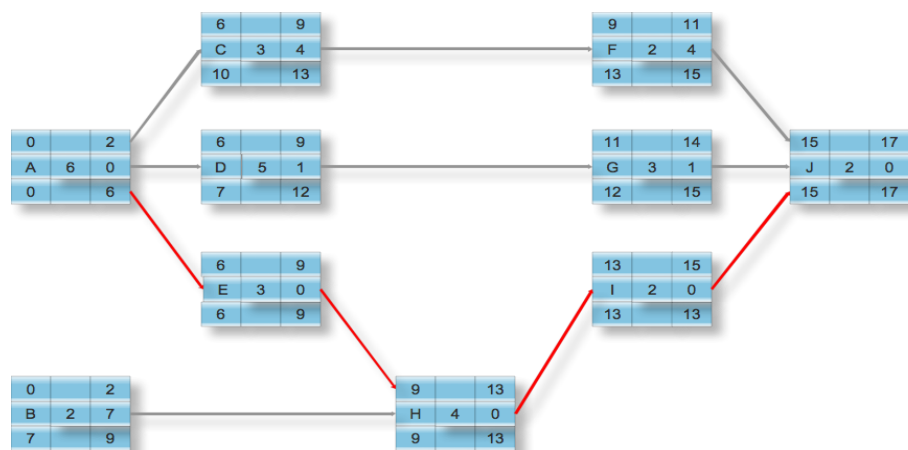
$$(1) \text{ Forventet varighet} = E(t) \frac{a + 4m + b}{36}$$

$$(2) \text{ Varians} = \text{Var}(t) = \frac{(b - a)^2}{6}$$

Aktiviteter	Optimistisk tid (a)	Mest sannsynlig tid (m)	Pessimistisk tid (b)	Forventet tid	Varians
A	4	5	12	$4+(4*5)+12/6 = 6$	$(12-4)^2/36 = 1,78$
B	1	1,5	5	$1+(4*1,5)+5/6 = 5$	$(5-1)^2/36 = 0,44$
C	2	3	4	$2+(4*3)+4/6 = 3$	$(4-2)^2/36 = 0,11$
D	3	4	11	$3+(4*4)+11/6 = 5$	$(11-3)^2/36 = 1,77$
E	2	3	4	$2+(4*3)+4/6 = 3$	$(4-2)^2/36 = 0,11$
F	1,5	2	2,5	$1,5+(4*2)+2,5/6 = 2$	$(2,5-1,5)^2/36 = 0,027$
G	1,5	3	4,5	$1,5+(4*3)+4,5/6 = 3$	$(4,5-1,5)^2/36 = 0,25$
H	2,5	3,5	7,5	$2,5+(4*3,5)+7,5/6 = 4$	$(7,5-2,5)^2/36 = 0,69$
I	1,5	2	2,5	$1+(4*2)+3/6 = 2$	$(2,5-1,5)^2/36 = 0,03$
J	1	2	3	$1+(4*2)+3/6 = 2$	$(3-1)^2/36 = 0,11$

Tabell 1 – PERT med usikkerhet (fritt etter Karlsen, 2013)

Tabell 1 er et eksempel på beregning av forventet tid og variansen ved å ta hensyn til tre estimater; optimistisk-, mest sannsynlig- og pessimistisk tid. Etter utregning av forventet varighet og variansen for aktivitetene A-J, slik det er vist ovenfor i tabellen, kan man tidsplanlegge det på akkurat samme måte som vi har vist tidligere under CPM. I figur 11 er eksemplet i tabell 1 illustrert ved AON – nettverk.



Figur 11 – AON –nettverk under PERT (fritt etter Karlsen, 2011)

Summen av stiene i figur 11:

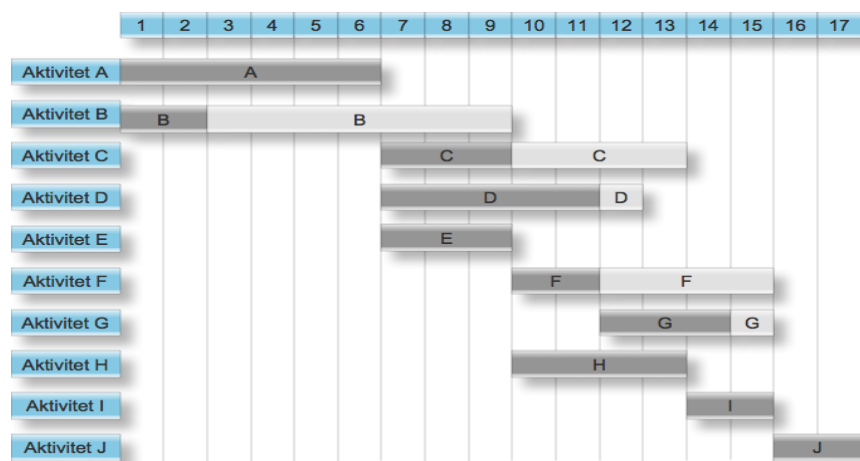
- A – C – F – J $6+3+2+2 = 13$
- A – D – G – J $6+5+3+2 = 16$
- A – E – H – I – J $6+3+4+2+2 = 17$
- B – H – I – J $2+6+2+2 = 12$

I eksemplet i figur 11 ser vi at den kritiske stien er på 17 tidsenheter. Den kritiske stien kan også beregnes ved å gå den motsatte veien av nettverket og se etter hvilke aktiviteter som kommer frem med null tidsenheter i flyt. Stien er markert med røde linjer mellom aktivitetene.

Man kan regne ut hva sannsynligheten er for at prosjektet blir fullført etter den tiden som er beregnet ut ved CPM. Dette gjøres ved å summere opp variansene til alle aktivitetene som er på kritisk sti ($1,78+0,11+0,69+0,03+0,11$). Det gir et standard avvik på 1,65 som vi kan sette i en formel med X som antall tidsenheter: $P(X \leq 20) = G \frac{20-17}{1,65} = G(1,82)$. Deretter brukes dette tallet til å slå opp i tabellen for normalkurven og se at det er 96% sannsynlighet for at aktivitetene blir gjennomført på 20 eller færre tidsenheter.

2.4.5 Gantt – diagram

Gantt – diagrammet er en annen måte å fremstille terminplanleggingen på. Her fremstilles de planlagte aktivitetene langs den ene aksen med tykke streker. Lengden på strekene angir varighetene på aktiviteten langs den andre aksen. Fordelene ved denne teknikken er at den er oversiktlig og lett å forstå. Aktivitetene kan markeres med forskjellige farger for å effektivisere bruken av diagrammet (Lock, 2013). Figur 12 er et eksempel på Gantt diagram som er utarbeidet ved å ta utgangspunkt i eksemplet i figur 11. Modellen viser aktivitetenes varighet (mørk grå streker) og aktivitetenes flyt (lys grå streker).



Figur 12 – Gantt- Diagram (fritt etter Karlsen, 2012)

Ulempen ved Gantt - diagrammer er at den ikke viser avhengighetsforholdene mellom aktivitetene og egner seg derfor ikke for praktisk styring. Det kreves at man på forhånd har tatt hensyn til avhengighetsforholdet, aktivitetens varighet og ikke minst disponering av ressurser før man plasserer aktivitetene på Gantt – diagrammet. Dette vil kanskje ikke være et problem under sekvensielle beslutninger (Rolstadås, 2011). Dersom man har et veldig enkel plan bestående av få aktiviteter kan avhengighetsforholdene fremstilles ved å tegne linker mellom aktivitetene. Dette blir kalt for linket Gantt – diagram (Lock, 2013).

2.4.6 Kostnadsplanlegging

Kostnadsestimering og budsjettering er nødvendig for å vite hva prosjektet vil koste og hvilke økonomiske rammer man skal forholde seg til. Kostnader er knyttet blant annet til ressurser og arbeidstid. I følge Karlsen (2013) kan kostnader defineres ved:

”...kostnader er en ressurs som ofres for å oppnå et bestemt mål”

Estimerings prosesser

Kostnader i et prosjekt kommer i form av betaling for arbeidskraft, utstyr eller materialer, og kan styres ved hjelp av fire prosesser (Karlsen, 2012):

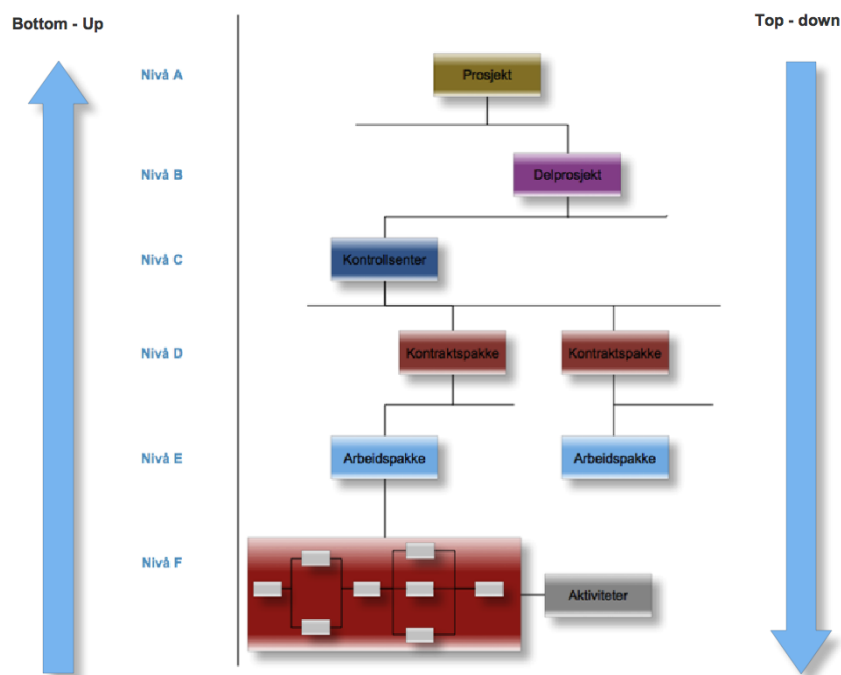
- 1) *Ressurserestimering* går ut på bestemme hvilke og hvor mye ressurser (arbeidskraft, utstyr, materialer og kapital) som trengs for å gjennomføre prosjektet i henhold til planen.
- 2) *Kostnadsestimering* går ut på å estimere kostnader på ulike ressursene man trenger i et prosjekt.
- 3) *Strukturering av kostnadene* går ut på å allokere kostnadsestimatene til individuelle arbeidsoppgavene for å etablere en ”baseline”.
- 4) *Kostnadskontroll* er en oppfølgingsprosess som går ut på å registrere avvik dersom faktiske kostnader avviker fra den budsjetterte planen. Prosessene bør holdes oppdaterte for at kostnadsplanen skal bli mest mulig nøyaktig og riktig.

Estimeringsteknikker

Det finnes flere estimeringsteknikker for kostnadsestimering. Vi beskriver her to prinsipper (Karlsen, 2012):

1. *Bottom – up* prinsippet går ut på at ressursbehovet og tilhørende kostnader blir estimert på detaljnivå i prosjektets WBS, og summert oppover til en total kostnad for hele prosjektet. *Fordelene* ved dette prinsippet er at ressursbehovet og estimater blir beregnet av dem som skal utføre prosjektarbeidet senere (realistiske). *Ulempen* ved dette kan være kunstige høye estimater som fører til at man mister troverdigheten.
2. *Top – down* prinsippet går derimot ut på at en prosjektleder får tildelt en økonomisk ramme som han/hun skal gjennomføre prosjektet for. Dette beløpet splittes deretter opp og allokeres til delprosjekter, arbeidspakker og aktiviteter. *Fordelene* er at det kan oppstå konkurranse mellom delprosjektledere om hvem som har høyest effektivitet og produktivitet og dermed skape mer verdi av tildelte midler. Den er svært enkel og krever lite ressurser. *Ulempen* kan oppstå hvis beløpet er tatt ut av luften. Det kan medføre til prosjektresultatet ikke svarer til det oppdragsgiveren forventer/ønsker.

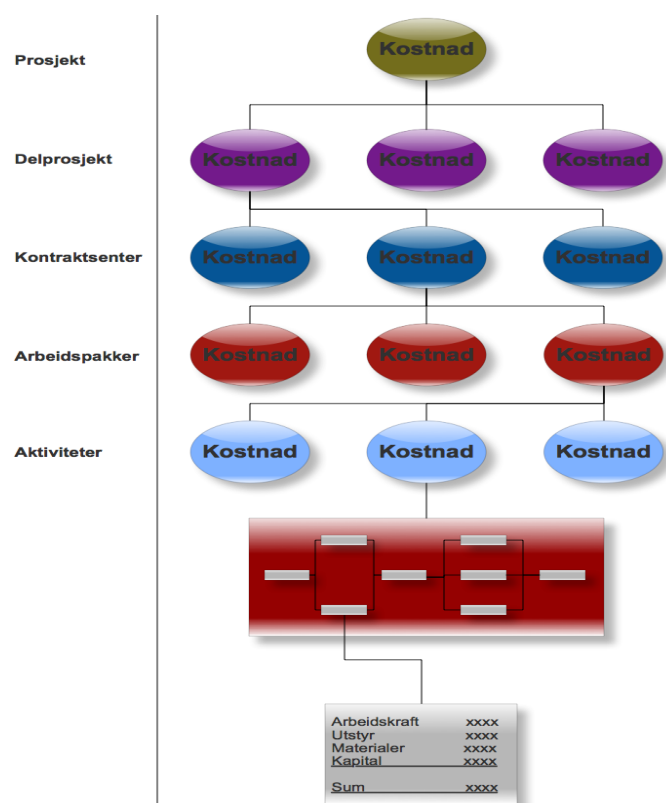
Figur 13 illustrerer WBS med estimeringsteknikkene top – down og bottom- up.



Figur 13 – Bottom – up og Top – down (fritt etter Karlsen, 2013)

CBS – kostnadsnedbrytning

Man kan strukturere kostnadene ved å benytte seg av strukturen Cost Break Down Structure (CBS) som i prinsipp nesten er identisk med WBS. Denne strukturen skiller seg ved at man her plasserer kostnadsinformasjonen istedenfor leveranser/funksjoner. Figur 14 er et eksempel på CBS.



Figur 14 – Cost Break Structure (Karlsen, 2012)

Direkte – og indirekte kostnader

Direkte kostnader er kostnader som knyttes direkte til prosjektarbeidet og gjennomføringen av aktiviteter (lønn, materiell kostnader, utstyr). I den ideelle situasjonen er direkte kostnader lik det vi kaller normale kostnader, det vil si lavest mulig kostnad som påløper for å gjennomføre en aktivitet så effektivt som mulig ved bruk av normal tid (Karlsen, 2012). Når prosjekttiden reduseres vil direkte kostnadene økes.

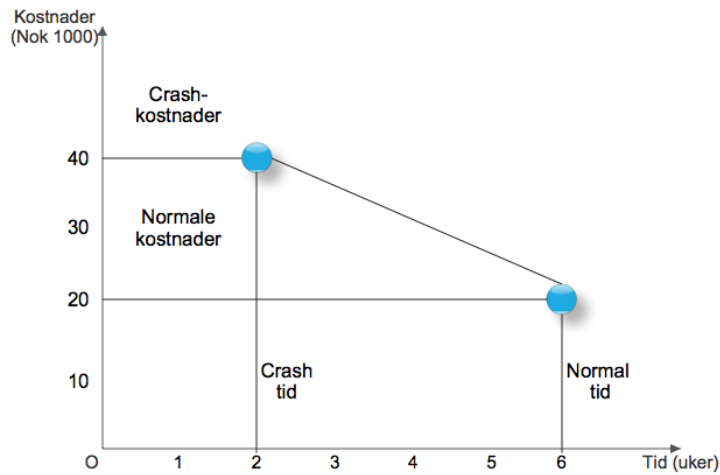
Indirekte kostnader er kostnader som påløper uansett hvilke arbeidsoppgaver som gjennomføres. Kostnader som er nødvendige for å holde virksomheten gående (utgifter til kopimaskin, sjefens lønn, forsikringer og rengjøringer). Disse er knyttet direkte til tiden, og vil derfor reduseres ved reduisering av prosjektiden (Karlsen, 2012). Tabell 2 viser hvordan direkte- og indirekte kostnader skiller seg fra hverandre.

Direkte kostnader (kan spores direkte)	Indirekte kostnader (kan ikke spores tilbake til enkelt aktivitetene)
<ul style="list-style-type: none"> • Arbeidskraft • Materialer • Utstyr • Reiser 	<ul style="list-style-type: none"> • Ledelse • Kontorleie • Strøm og forsikring • Rengjøring

Tabell 2 – Komponenter i kostnadsestimat og budsjett

Ved summering av direkte- og indirekte kostnadene får vi totale kostnader, og kan dermed se den optimale punktet mellom kostnader og tid. Dersom prosjektets varighet skal reduseres må vi finne ut hvilke aktiviteter på kritisk sti kan akselereres. Deretter kan vi beregne kostnad pr. tidsenhet for kritiske aktivitetene og sammenligne dets kostnader. Følgende begreper brukes ved sammenligning av aktivitetenes kostnader og tid (Karlsen, 2012):

- *Normal tid*: varighet som følger ved optimal planlegging hvor hver aktivitet gjennomføres på en normal måte
- *Crash tid*: varighet som følger av at aktivitetene gjennomføres så raskt som teknisk mulig. Crashtid vil alltid være kortere enn normal tid.
- *Normale kostnader*: Minimum direkte kostnader for å utføre aktivitetene på en rasjonell måte ved bruk av normal tid.
- *Crash kostnader*: Minimum direkte kostnader for å klare crash tiden. Crash kostnader vil alltid være større enn normale kostnader.



Figur 15 – Crashing av en aktivitet (fritt etter Karlsen, 2012)

Figur 15 gir et eksempel av en aktivitet med normal tid på 6 uker som kan reduseres med 4 uker ved at man øker kostnader med 20 000 kroner.

$$\text{Kostnad pr. tidsenhet} = \frac{40\,000 \text{ (Crashkostnader)} - 20\,000 \text{ (Normale kostnader)}}{6 \text{ uker (Normal tid)} - 2 \text{ uker (Crashtid)}}$$

Kostnad per tidsenhet for denne aktiviteten er 5000 kroner som da kan sammenlignes med andre aktiviteter som også befinner seg på den kritiske stien. Deretter velger man aktiviteten med lavest kostnad per tidsenhet.

2.4.7 Ressursplanlegging

Ressursplanlegging handler om å utarbeide en plan ut fra tilgjengelige ressurser slik at prosjektet får mest mulig igjen for ressursene (Karlsen, 2012). Gjennomføring av flere prosjekter samtidig, vil føre til at prosjektene må konkurrere om ressursene. Dette fører videre til forsinkelser av aktiviteter og arbeidsoppgaver.

Ressurser

Som oftest er det begrensninger på ressursene man trenger i et prosjekt og er derfor nødvendig med å utarbeide en plan som optimaliserer bruken av tilgjengelige ressurser.

Nedenfor er det listet opp generelle ressurser med dets begrensninger (Karlsen, 2012):

- Arbeidskraft – menneskelige ressurser utgjør som oftest kompetanse og hvor noen kan være vanskeligere å skaffe enn andre.
- Materiell – tilgangen til materielle ressurser er ofte en risiko og har forsinket mange prosjekter. Anskaffelse av materiell bør derfor planlegges så tidlig som mulig
- Utstyr – dette kan være karakterisert ved type, størrelse og omfang. Godt utstyr kan bidra til å redusere prosjekttiden
- Arbeidskapital – mangel på arbeidskapital kan redusere prosjektleders handlefrihet til for eksempel å gjennomføre aktiviteter parallelt.

Ressursplanlegging ved tidsbegrenset prosjekt

Når ressursplanleggingen skal foretas ved et prosjekt må man ta hensyn til tidsbegrensningen og forholde seg til prosjektets levetid. Et virkemiddel for god kostnadsestimering for alle prosjekter er å optimalisere bruken av ressursene, og det gjelder både om prosjektet har begrenset eller ubegrenset tilgang til ressurser (Karlsen, 2012). Eksempelvis kan et prosjekt ha ubegrenset tilgang til ressurser med fastslått dato, hvor det må foretas ressursutjevning. Her kan det være viktig med følgende:

- Være forberedt på å ansette ekstra personell
- Tillate overtid
- Leie inn konsulenter
- Skaffe ekstra utstyr

I dette tilfellet vil det være to oppgaver som står sentralt (Karlsen, 2012):

- Beregne ressursbehovet som trengs for å gjennomføre aktivitetene (anvende mest mulig økonomisk bruk av ressursene).
- Tidsplanlegge aktivitetene slik at ressursbruken blir så jevn som mulig, samtidig som sluttdatoen til prosjektet ikke blir overskredet.

Dersom tilfellet er begrenset ressurstilgang med justerbar sluttdato, må ressursallokering planlegges nøye. I et ressursbegrenset prosjekt kan man ikke

overskride nivået på ressursuttak. Dersom det er mangel på ressurser kan forsinkelse aksepteres, men det avhenger av kundens meninger. I en situasjon som denne bør man først fremstille et ressursdiagram som viser ressursbelastningen over tid. Dette vil indikere overskridelser som ikke kan aksepteres. Man kan for eksempel forsøke å endre på starttidspunktene på aktiviteter som ikke befinner seg på en kritisk sti (starte senere i forhold til deres flyt), for det vil ikke påvirke prosjektets sluttdato. Man kan også endre startdatoene på kritiske aktiviteter, men dette vil føre til senere avslutning av prosjektet.

2.5 Planlegging i Agile modeller

Agile metoder hevdes å være en "ny" måte å tenke på i forhold til styring og gjennomføring av prosjekter. Det blir også sett på som et rammeverk for å ivareta uforutsigbarhet i prosjekter og håndtere endringer raskt, herunder løpende innspill fra kunde.

2.5.1 Dekomponering av produktkrav

For å planlegge release - og sprint planen er det nødvendig med å bryte ned produktkravene først slik tabell X nedenfor vises.

Requirement Level	Requirement
Theme	See account data with mobile application
Features	<ul style="list-style-type: none"> ➤ See account balances ➤ See a list of recent withdrawals and purchases ➤ See a list of recent deposits ➤ See upcoming automatic bill payments ➤ See my account alerts
Epic User Stories (decomposition for feature: see account balances)	<ul style="list-style-type: none"> • See checking account balance • See savings account balance • See investment account balance • See retirement account balance
User Stories (decomposition for epic user story: see checking account balance)	<ul style="list-style-type: none"> -Log into mobile account -Securely log into mobile account -See a list of my accounts -Select and view my checking account -See account balance changes after withdrawals and purchases -See day's end account balance -See mobile application navigation items -Change account view -Log out of mobile application

Tabell 3 – Dekomponering av produktkrav (Fritt etter Layton, 2012)

Tabell 3 er et eksempel på en mobilbankapplikasjon, hvor kontooversikt er et av produktkravene. Dette kan nedbrytes i funksjoner som gir oss en oversikt over hvilke funksjoner det utvalgte produktkravet skal inneholde. Deretter nedbrytes dette igjen i epic stories og user stories. Under sprint planning kan user stories igjen nedbrytes til oppgaver, slik det er vist i tabellen under.

Requirement Level	Requirement
Task (decomposition for User story: Log into mobile account)	<ul style="list-style-type: none"> ○ Create an authentication screen for a username and password, with a Submit button ○ Create an error screen for the user to re-enter credentials ○ Create a logged-in-screen (includes list of accounts – to be completed in next user story) ○ Using authentication code from the online banking application, rewrite code for an iPhone / iPad application ○ Create calls to the database to verify the username and password ○ Refactor code for mobile devices

Tabell 4 – Dekomponering av produktkrav vol. 2 (Fritt etter Layton, 2012)

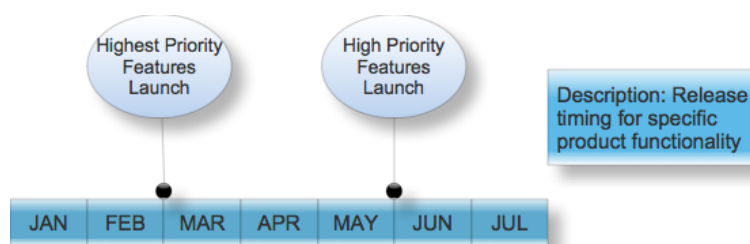
Smidig planlegging blir utført i to trinn for å støtte nødvendig fleksibilitet

(Szoke, 2011):

- *Releaseplan* - er en overordnet nivåplan som går ut på å vurdere et sett av funksjoner og ressurser (utviklere) som er med på å forme et produkt eller et delprodukt i en gitt leveranse tid.
- *Iterasjonplan* – er en mer detaljert nivåplan som går ut på å vurdere tekniske oppgaver (som er avledet fra funksjoner), ressurser (utviklere) og leveranse stadiene.

2.5.2 Releaseplan

Releaseplan kan oversettes til lanseringsplanen på norsk som blir en overordnet veikart for å komme fram til målet. En releaseplan er en overordnet nivåplan som går ut på å vurdere et sett av funksjoner og ressurser (utviklere) som er med på å forme et produkt eller et delprodukt i en gitt leveranse tid. Denne releaseplanen kan bestå av flere iterasjoner som kan være på fra en uke og opptil 4 uker. Figur 16 er et eksempel på releaseplaner; høyst prioritert og høy prioritert leveranser.



Figur 16 – Releaseplan (Layton, 2012)

Releaseplanleggings trinnet består av to hovedprosesser:

- Produktkrav utvalg
- Produktkrav tidsestimering

Utvalg- og tidsestimeringsprosessen av et krav er ofte gjort iterativt for å finne gruppe av aktiviteter for å nå ”on time delivery project plan”.

Krav utvalg

Ofte er det flere krav enn det som kan bli implementert og er derfor viktig at man prioriterer kravene i riktig rekkefølge. Kravlisten må prioriteres i henhold til interessene til ulike interessentene og ta andre variabler i betraktning som for eksempel forretningsverdi, interessent preferanse, utviklingskostnaden osv. Deretter redegjør man funksjonens avhengighetsforhold med andre funksjoner ved å tildele et av de fem avhengighetsforholdene (Li m.fl 2010):

- Kombinasjon: to funksjoner skal gjennomføres i fellesskap
- Implikasjon: en funksjon krever at en annen skal fungere
- Eksklusjon: to funksjoner i konflikt med hverandre

- Tidsrelatert: en funksjon er avhengig av at en annen funksjon er implementert.
- Omsetningsbasert og kostnadsbasert: en funksjon påvirker inntektene/kostnadene for en annen funksjon.

Dependency group	Dependency type	Influence requirement selection	Influence requirement scheduling
Functional dependency	Combination	✓	
	Implication	✓	✓
	Exclusion	✓	
Value – related dependency	Revenue- based	✓	
	Cost- based	✓	✓
Time – related dependency	Time - related		✓

Tabell 5 – Avhengighetsforholdene (Li. m.fl (2010))

Krav tidsestimering

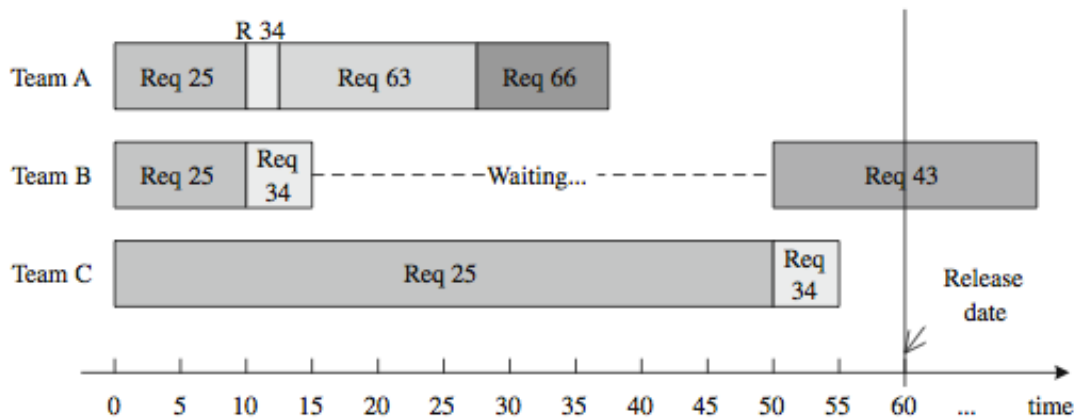
Etter kravutvalg kan man starte med det andre steget, krav tidsestimering. I tabell 6 er det listet opp ni produktkrav med dets omsetning, kostnad (man days), avhengighetsforhold og utviklingsteamets kapasitet ved hjelp av Integer linear programming (ILP) teknikken. ILP er videreutviklet med genetisk algoritme (Li m.fl, 2010). Genetiske algoritmer er en løsning for å avgjøre avveining for releaseplaner ved å tillate variasjonen av releasedatoer og/eller ressurs kapasiteter innenfor forhåndsdefinerte intervaller på gjennomførbarhet i forbindelse med et definert mål på risiko (Elroy and Ruhe, 2010).

Nr.	Requirement	Preceded by	Revenue	Total man days	Team A	Team B	Team C
12	Authorization on order cancellation and removal	25,63	192	400	40	0	360
34	Authorization on archiving service orders		96	96	16	40	40
63	Performance improvements order processing		160	120	120	0	0
25	Inclusion graphical plan board		800	560	80	80	400
43	Link with Acrobat reader for PDF files	25	80	264	0	264	0
75	Optimizing interface with international Postal code system	25	80	120	0	0	120
35	Adaptations in rental and systems		280	320	0	160	160
66	Symbol import		40	80	80	0	0
67	Comparison of services per department		80	272	0	72	200
Total			1808	2232	336	616	1280
Available resources (number of developers)				24	8	8	8
Available team capacity for release				1440	480	480	480
Release duration					60 days		

Tabell 6 – Eksempel på et krav- og releaseplan problem (Li m.fl, 2010)

Tabell 6 er et eksempel på en releaseplan som består av totalt ni produktkrav som ønskes å utvikles i en releaseplan på 60 dager. Alle Utviklingsteamene A, B og C har en kapasitet på 480 man days. Utviklingsteamet B og C er overbelastet med 616 og 1280 man days da de kun har kapasitet til 480 hver. Utviklingsteamet A har fortsatt kapasitet til å ta inn mer jobb da de sitter med 144 man days til gode ($480 - 336 = 144$), men siden hver utviklingsteam har sin spesielle kunnskap er det begrensninger med hva de kan bidra med.

For å løse ressursproblemet er fem av produktkravene valgt ut for en release plan med en varighet på 60 dager. Produktkravet nr.43 har en avhengighets forhold og kan derfor starte etter at produktkravet nr. 25 er ferdig. Andre produktkravene kan jobbes parallelt med andre kravene av utviklingsteamene. Modellen i figur 17 illustrerer tidsplanen for overnevnte eksempel.



Figur 17 – Tall eksempel på et krav tidsestimerings problem (Li m.fl, 2010)

Det første man noterer i figuren over er at prosjektet ikke vil holde releasefristen på 60 dager. Årsaken til det er at det foreligger en implikasjons avhengighet og dermed en presedensbegrensning mellom produktkravet 25 og 43. Team B kan starte med produktkravet 43 etter at produktkravet 25 er ferdig. Selv om team B er ferdig med kravet 25 på dag 10 så må de vente til team C er ferdig. Team C er ferdig med produktkravet på dag 50 og dermed kan Team B starte med produktkravet 43 som er estimert til 33 dager. Team B må vente i 35 dager som fører til at prosjektet blir forsinket med 23 dager ($50 + 33 = 83 - 60 = 23$). Vi ser at presedens forholdet påvirker tidsplanen som stemmer overens med tabell 5.

Det er nettopp slike problemer man ønsker å løse ved hjelp av prosjektstyringen. Her er det nødvendig med replanlegging av hvilke produktkrav som skal tas med for å holde tidsfristen på 60 dager. Produktkravet 43 må enten tas bort fra releaseplanen eller erstattes med andre produktkrav som kan utvikles innenfor tidsfristen.

2.5.3 Iterasjon plan

Etter at produktkravene er valgt ut ved å ta hensyn til avhengighetsforholdene kan iterasjon planleggingen startes. I dette steget nedbrytes produktkravene til oppgaver (se tabell 4). Deretter estimeres oppgavene i timer som gjennomføres i flere iterasjoner for å redusere den overordna risikoen ved produktutviklingen (Szokes, 2011).

Gradering av oppgaver

Oppgavene i user stories estimeres i timer. Dette starter med å gradere oppgavene som har to hensikter; den ene er å kunne rangere oppgavene etter prioritering, den andre er å gjøre forenkle oppfølgingen og å kunne estimere fremdriften (Pichler 2010). Det finnes to forskjellige måter å gradere oppgaver på; Story point og planning poker. De kan fungerer hver for seg, eller Planning poker kan bygge videre på Story points. Gradering av oppgavene i Scrum kan gjøres på en grov overordnet måte på produkt backlogg nivå, eller på en detaljert måte på sprint nivå. Videre skal vi ta for oss redegjørelsen på graderingen for produkt backlogg nivå.

Story Points

Story point bygger på grove estimeringer av hastighet (velocity) og størrelse, og brukes til å indikere kompleksiteten i en oppgave. Gir man en oppgave story point to er den dobbel så stor som en oppgave med story point en. En oppgave som er verdt story point 3 skal være så mye som story point en og to sammenlagt. Hvor store disse størrelsene er tolkes relativt, det går ut på hva som er utgangspunktet for målingen. Som når man sammenligner datamusen i forhold til dataskjermen, blir datamusen oppfattet som forholdsvis liten, men den er stor sammenlignet med en usb-stick. I tabell 7 finner man en rangering av Story points som er mye brukt.

Story Point	T-skjorte størrelse
0	Freebie, oppgaven har allerede blitt implementert
1	XS - Extra small
2	S - Small
3	M - Medium
5	L - Large
8	XL - Extra- large
13	XXL - Double extra-large
20	XXXL - Huge

Tabell 7 - Story points rangering (Layton, 2012)

For å fremskynde teamets beslutningsprosess er rangeringen på tabell 7 en ikke-lineær sekvens. Meningen med det er å forhindre langvarige diskusjoner som kan oppstå dersom det var en sekvensiell rangering. Om ønskelig er det mulig å ekspandere rangeringen med 40 og 100 for å ytterligere markere kompleksiteten, så lenge den relative estimeringen er korrekt (Pichler, 2010).

Planning poker

Planning poker er en effektiv team-basert estimeringsteknikk for kompleksitet på en oppgave. Den kan brukes alene eller bygge på story point. Til å begynne med lager teamet en kortstokk. Innholdet på kortene baserer seg enten på estimerer bestemt av teamet eller på story point rangeringen det er utarbeidet. Eksempelvis gir rangeringen på tabell 7 8 kort. Kun utviklere av utviklingsteamet skal være med og kaste ut sine kort. Med andre ord utelates produkteier og Scrummaster. Kaster alle ut like kort blir oppgaven enstemmig estimert. Er det noen som kaster ut forskjellige kort må laveste og høyeste tall forsvare sitt valg med en kort begrunnelse.

I et nytt prosjekt, hvor ingen utviklere har erfaring med oppgavene som skal utvikles, begynner man med å estimere den enkleste oppgaven, for så deretter estimere alle andre oppgaver etter den. Har noen utviklere i teamet derimot erfaring med oppgavene som skal utvikles og kjenner til kompleksiteten på dem begynner man med den mest komplekse oppgaven, for deretter å estimere seg nedover. En slik estimeringsrunde kan gjerne ta fra en halv- til en heldagsarbeidstid.

2.6 Oppfølging i tradisjonelle metoder

Statusrapporter utarbeides for å kontrollere om prosjektet går etter planen eller ikke. Dette kan for eksempelvis gjøres ved å sammenligne budsjettert plan med den faktiske situasjonen, holde systematiske oppfølgingsmøter eller ved å ha samtaler med prosjektmedlemmene.

Oppfølgingskriterier

Oppfølgingskriteriene gjenspeiler prosjektets ”NÅ” situasjon på detaljnivå. En statusrapport bør derfor inneholde følgende forhold (Karlsen, 2013):

- *Kostnader (ressurser)* – gir en oversikt over påløpte kostnader/ressurser hittil i prosjektet. Deretter se hva som gjenstår av behov for kostnader/ressurser, slik at vi har en sluttsum (påløpte + gjenstående = sum) som kan sammenlignes med budsjettert plan
- *Tidsplanen* - er en oversikt om hvorvidt prosjektet holder tidsplanen ved for eksempel måle fremdriften ved å følge den kritiske linjen. Hvis den planlagte og den faktiske verdien er den samme og det likevel er rapportert om forsinkelser er det et tegn på at man har jobbet med aktiviteter som ikke ligger på den kritiske linjen.
- *Kvaliteten* – går ut på å sikre prosjektresultatet oppfyller de kravene oppdragsgiveren har satt.
- *Omfang* – gir er en oversikt over eventuelle endringer i planen.
- *Usikkerhet* – blir rapportert for å unngå og/eller for å være forberedt på uønskede hendelser i prosjektet.
- *Ventetid* – det blir redegjort for å se om det har vært ventetid for prosjektmedarbeidere. Ventetid anses som ressursløsning, da et prosjekt kan ha ressursbegrensninger.

Hensikten med oppfølging

Hensikten med oppfølgingen av kriteriene er å få et grunnlag for å avgjøre om det eventuelt er behov for korrigerende tiltak. Et eksempel på dette er om prosjektet ikke ligger etter den opprinnelige planen. De vanligste tilfellene en står ovenfor er negativt avvik i tid og/eller i kostnader (Karlsen, 2013). Hvis det fremkommer i rapportene om at det er svikt i framdriften kan følgende beslutninger tas (Andersen, Grud og Haug, 2009):

- Flytte milepælen(e) i tid
- Senke ambisjonsnivået
- Tilføre ytterligere ressurser
- Omfordele arbeidet
- Fortsette som før

Uavhengig av beslutningsvalg bør planen replanlegges for å finne den nye kritiske veien. Årsaken til det siste beslutningsalternativet (fortsette som før) kan være en sterk tro på at det oppståtte problemet vil gå over av seg selv og at det mest sannsynlig vil føre til kostnadsoverskridelser (Karlsen, 2013). Dette alternativet går mot holdninger som ikke er tillatt som for eksempel ”det går nok” eller ”vi får vente og se” (Andersen, Grud og Haug, 2009). Vi vil konkludere dette med at dersom det finnes tiltak for svikt i framdriften bør man iverksette dette umiddelbart og ikke ta sjansen med å vente, som igjen krever replanlegging.

2.6.1 S-kurver - Integrert KTR-styring

S – kurve er et hjelpemiddel i oppfølgingsprosessen som viser akkumulerte kostnader eller arbeidstimer (Lock, 2013). Kurven får en S – form da start- og sluttkostnadene er som regel lavest i begynnelsen av et prosjekt, øker i gjennomføringsfasen og avtar når prosjektet går mot avslutning. S-kurven vil være forskjøvet mot venstre eller høyere avhengig av om aktivitetene er orientert mot *tidligste start* eller *seneste start (CPM)*. Det beregnes tre forskjellige S – kurver i samme diagrammet som er (Lock, 2013):

- *Planlagt verdi* – kostnader som er planlagt brukt i prosjektet. **BCWS** – Budgeted Cost of Work Scheduled.
- *Virkelig kostnader* – som har påløpt til enhver tid for å utføre de planlagte aktivitetene. **ACWP** – Actual Cost of Work Performed.
- *Inntjent verdi* – det utgjør verdiskapningen av utført arbeid i prosjektet. Dette beregnes ved å multiplisere ”prosent av arbeid fullført” med planlagt budsjett ved fullføring BAC (budget at completion). **BCWP** – Budgeted Cost of Work Performed.

Ved hjelp av S-kurvene kan man fremskaffe viktig styringsinformasjon som (Karlsen, 2012):

Kostnadsavvik

Viser forskjellen mellom estimert kostnad og virkelig kostnad for en aktivitet. Et negativt tall betyr at utførelsen koster mer enn planlagt.

- $CV \text{ (cost variation)} = BCWP - ACWP$

Budsjettavvik

Dette avviket beregnes ved å ta planlagt verdi minus kostnader. Et negativt avvik betyr at prosjektet har brukt mer penger enn budsjettet hittil.

- $BV \text{ (budget variation)} = BCWS - ACWP$

Fremdriftsavvik

Viser forskjellen mellom en planlagt ferdigstillelse av en aktivitet og virkelig ferdigstillelse av aktiviteten. Et negativt tall betyr at arbeidet tok lenger tid enn planlagt å fullføre.

- $SV \text{ (scheduled variance)} = BCWP - BCWS$

En annen måte å fremskaffe styringsinformasjon på, er å beregne indekser basert på inntjent verdi (Karlsen, 2012):

Produktivitetsindeks

Cost Performance Index (CPI) – beskriver hvor kostnadseffektivt og produktivt prosjektet har vært i det arbeidet som er utført. CPI er forholdet mellom inntjent verdi og virkelig kostnad (BCWP / ACWP).

- Hvis CPI er større enn 1, gir det informasjon om at produktiviteten er god. Fortsetter denne utviklingen, vil prosjektets totale kostnader bli lavere enn budsjettet.
- Hvis CPI er mindre enn 1, gir det informasjon om at det er sannsynlig at prosjektet vil overskride budsjettet.

Fremdriftsindeks

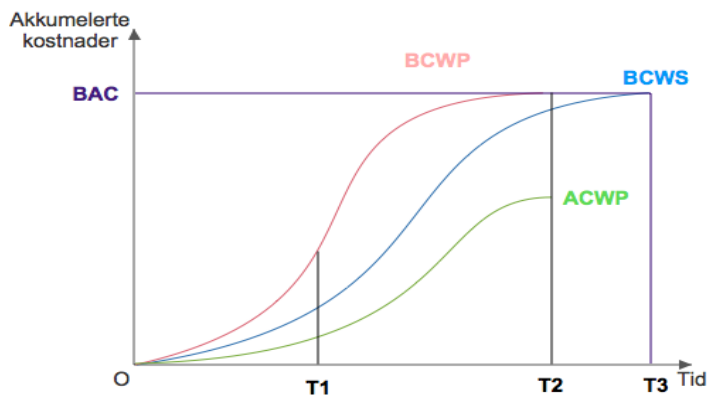
Schedule Performance Index (SPI) – Sier noe om prosjektet holder tidsplanen eller ikke. SPI er definert som forholdet mellom inntjent verdi og planlagt verdi (BCWP / BCWS).

- Hvis SPI er større enn 1, gir dette en indikasjon på at prosjektet vil bli gjennomført på kortere tid enn planlagt.
- Hvis SPI er lavere enn 1, betyr det at det har oppstått en forsinkelse, d.v.s. at prosjektet ikke klarer å holde tidsplanen.

Estimate at Completion - EAC

Ved å ta utgangspunkt i CPI kan prosjektets sluttkostnad også beregnes.

- $EAC = ACWP + ((BAC - BCWP) / CPI)$
- Sluttid $T = \text{planlagt totaltid} / SPI$



Figur 18 - S – Kurven (fritt etter Karlsen, 2013)

Figur 18 er et eksempel på en S-kurve som har en positiv utvikling da faktiske kostnader, ACWP, er lavere enn budsjetterte, BCWS. I følge figur 14 kan dette føre til at inntjent verdien, BCWP, er mye høyere enn BCWS. Forutsetningen for at inntjent verdi er større enn de faktiske kostnader er at effektiviteten, CPI og produktiviteten, SPI er større enn 1,00. Forutsetningen for at inntjent verdi er større enn planlagt verdi, er at mer enn det planlagte arbeidet i kontrollpunktet er unnagjort. Det vi også kan se ut fra modellen er at prosjektet mest sannsynlig vil bli ferdig i tidspunkt T2 med lavere kostnader BAC, og ikke T3 som var planlagt.

Reduksjon av prosjekttiden

Tid er en verdi som ofte anses i form av penger. Fordelene ved tidlig ferdigstillelse av prosjekter kan være (Karlsen, 2012):

- gir raskere gevinster og resultater
- forsprang på sine konkurrenter
- frigir prosjektmedarbeidere og ressurser tidligere enn planlagt (nye oppdrag tidligere)
- fordi man på et tidlig tidspunkt, før planer er utarbeidet, gått ut og kunngjort en sluttdato. (ny teknologi)

Det kan også foreligge andre årsaker for å redusere prosjekttiden. Eksempelvis kan det være at et prosjekt er forsinket og man må korte ned prosjekttiden for å få levert produktet i tide. Prosjekttiden kan reduseres ved å ta utgangspunktet i aktivitetene på den kritiske stien, da det er kritiske stien gjennom nettverket som bestemmer

prosjektets varighet. Raskere gjennomføring av aktiviteter som har flyt, gir ingen effekt på sluttiden. Reduksjon av prosjektets varighet kan skje ved å (Karlsen, 2013):

- gjennomføre arbeidsoppgaver i parallell om det er mulig (krever mer koordineringer enn ved sekvensiell)
- sette bort arbeidsoppgaver til underleverandører (krever forarbeid og anbudsrunder for kostnadsbesparelse)
- redusere kvaliteten (oppdragsgiveren vil nekte dette)
- redusere omfanget (utelate funksjoner som ikke har store konsekvenser)
- tilføre ekstra ressurser i form av personell og utstyr (crashing – vil føre til ekstra kostnader)

2.7 Oppfølging i Agile Metoder

En av hovedfordelene med Scrums styringsmodellen er at oppfølgingen av prosjektet skjer fortløpende hver dag gjennom daglig Scrum møter og ved sprint evalueringer etter endt sprint. Deretter oppdateres produkt- og sprintbackloggen. Fremgangen i utviklingen i et prosjekt holdes oversiktlig ved styringsverktøyet release burndown.

2.7.1 Release burndown

Et styringsverktøy fram mot releasedatoen eller for å holde orden på sprintene er å bruke releaseburndown. Den bruker hastigheten på oppgavene, hvor mye tid som er igjen og hvor mye tid som er brukt. Burndown vil si hvor mye tid som er forbrent på hver oppgave, altså hvor mye tid som er brukt. Det tillater oss å spore og estimere framgangsprosessen i prosjektet (Pichler 2010). Verktøyet kan benyttes i oppfølgingen av både release og sprint (iterasjon).

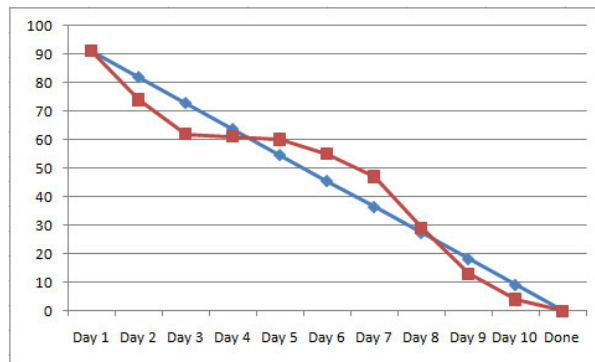
Det finnes to forskjellige burndown visninger, burndown chart og burndown bar:

Burndown Chart

Denne viser oss som sagt fremgangen i prosjektet i et diagram ved å basere seg på 2 følgende faktorer: den gjenstående innsatsen i produkt backloggen, og tid. Den kan

dessuten beregne fremtiden for prosjektet slik at teamet kan tilpasse produktet og prosjektet når det trengs basert på hastighetene av de forrige sprintene¹.

Etter et sprint-review møte er sprintens resultat kjent, det er da den beste tiden for å lage og oppdatere burndown diagrammet. Figur 19 er et eksempel av burndown diagrammet.



Figur 19 – Burndown Chart (link 4)

Grafen i figur 19 viser ”gjenstående innsats” på y-aksen, mens tid vises på x-aksen. Den blå linjen er referanse linjen og viser det ”normale” eller standarden for en lineær burndown. Diagrammet har alltid en slik linje slik at man har noe å sammenligne med og se hvor mye man avviker fra standarden. Den røde linjen viser den virkelige burndown. Når man lager dette diagrammet har man den friheten til å tilpasse den etter organisasjonens eller teamets behov; ”gjenstående innsats” kan eksempelvis settes til dager, timer, uker eller Storypoints, og tid kan settes til dager eller uker.

Burndown bar

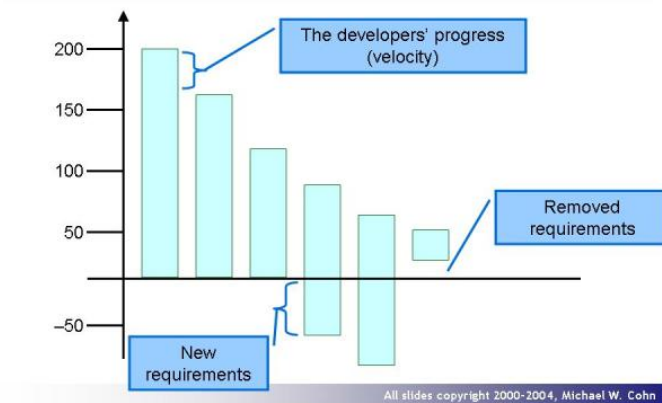
I følge Cohn er dette en mer sofistikert versjon av burndown diagrammet (Cohn 2005,221-24). Burndown bar er annerledes i den grad at den reestimerer elementer og brenning innsats på den ene siden og legger til og fjerner backlogg elementer på den andre siden. Toppen av baren vil gå ned dersom teamet gjør fremgang eller reduserer tidsestimatene. Om teamet øker estimatene, vil toppen av baren gå oppover.

Om nye elementer bli lagt til i produkt backloggen vil bunnen av baren gå nedover.

Men dersom elementene blir tatt ut av backloggen eller avløst med noe som krever

mindre hastighet, vil bunnen gå opp. En illustrasjon av burndown baren kan se slik ut:

A more powerful release burndown



Figur 20 – Burndown Bar (Cohn, 2005)(Link 5)

Figur 20 (Cohn, 2005) viser at barene går nedover når nye elementer blir lagt til, og elementene som blir tatt vekk forsvinner fra bunnen av barene. Da kan de bli så korte at de ser ut som de ”flyter” over x-aksen. Y-aksen kan betegnes med eksempelvis Story points og X-aksen kan være antall sprinter.

2.7.2 Velocity

Velocity eller hastighet på norsk, er indikatoren på summen av innsats for arbeidets resultat akseptert av produkteier i en sprint. Det viser hvor mye arbeid et team kan gjøre i en sprint og det tillater oss å spore og forutse prosjektets framgang.

Hastigheten, som vises i form av tid, blir fastslått av teamets evne til å utføre oppgaver fra produktbackloggen om produktets tilvekster. Avhengig av teambuilding dynamikker, hindringer, eller tilgjengelighet kan hastigheten variere. Man kan som regel ikke sammenligne hastigheten mellom team i et prosjekt, hvis ikke teamene bruker story points med den samme meningen. Å vite at team 1 har en hastighet på 40 på å utvikle produkt A og team 2 har en hastighet på 20 på å utvikle produkt B betyr ikke at team 1 er bedre enn team 2. Det kan være at produkt A har lavere estimater (Pichler,2010). I et team hvor medlemmene aldri har jobbet sammen før, estimerer man gjerne hastigheten etter at sprint 1 er ferdig, for å se hvordan samarbeidet påvirker hastigheten. Deretter regner man det ut for de resterende sprintene. Estimaten blir gjerne bedre etter at man har gått gjennom 2 eller 3 sprinter.

Kapittel 3 – Metode

I dette kapitlet beskriver vi kvantitative og kvalitative metoder med deres fordeler og ulemper. Deretter redegjør vi for vårt valg av metode samt krav til bruk av metode og innsamlede data.

3.1 Innledning

Metode kan defineres som en planmessig fremgangsmåte som avhenger av vårt mål (Dalland, 2007). Dette innebærer en beskrivelse av hvordan datainnsamlingen, analyse av data, tolkning og ikke minst rapportering skal gjøres. Dette er altså et redskap som man benytter seg av når man ønsker å undersøke en hendelse eller et fenomen. Ved bruk av dette redskapet vil man kunne dokumentere de valg man har tatt og vise hvordan resultatene er fremkommet for andre.

3.2 Kvantitative og kvalitative metoder

Metoder dataene samles inn på kan karakteriseres som enten kvalitative eller kvantitative. Kvantitative metode er en metode hvor data kan omformes og uttrykkes i tall eller mengdeenheter. Det gjør det mulig å beskrive et fenomen ved hjelp av tall. Nøyaktighet er et sentralt stikkord når man ønsker å veie kvantitative data. Kvantitative data har da sin styrke for når man ønsker å svare på spørsmålet som ”hvor mange/mye”. Ved kvantitative metoder har forskeren stor avstand til undersøkelsesenheter og forholder seg som regel til flere enheter.

Kvalitative data er data som ikke kan tallfestes. I kvalitativ analyse er man opptatt av fullstendigheten enn nøyaktighet. Dataen har derfor sin styrke når man ønsker å besvare spørsmålet som hva, hvorfor og hvordan. Her dreier det seg mer om å forstå enn å måle. Derfor er det viktig at forskeren forstår og tolker riktig.

Kvalitative metoder blir tatt i betraktning hvor man ønsker å få kunnskap om menneskers virkelighetsforståelse, deres tanker, erfaringer og meninger. Datainnsamlingen i denne metoden foregår ved deltagende observasjon, diskurs/tekstanalyser eller intervjuer. I tabellen under har vi oppsummert hvordan kvantitative og kvalitative metoder skiller seg ved ulike fasene.

	Kvalitative	Kvantitative
Mål	Forståelse	Tallfeste, generalisere
Utvalg	Liten	Representativ
Datainnsamling	Ustrukturert	Strukturert
Dataanalyse	Ikke statistisk	Statistisk
Utbytte	Førstehånds forståelse	Foreslå handlingsalternativer

Tabell 8 – Kvalitativ VS Kvantitativ

Det finnes både fordeler og ulemper ved kvalitative og kvantitative metoder (Gripsrud, 2004).

Hovedsakelig skjer datainnsamlingen gjennom intervjuer i kvalitative metoder som er med på å gi oss flere fordeler. *Fordeler* ved kvalitative metodene er at man kan få avstemt en ide med noen nøkkelpersoner tidlig i en prosess og ikke minst fleksibilitet da man har stor medvirkning under intervjuet.

Det finnes en rekke *ulemper* ved kvalitative metoder. Man kan ikke generalisere eller tallfeste resultatene, resultatene vil aldri være representative, det kan være vanskelig å styre samtalene, tolkning tar lang tid, resultatene kan vanskelig etterprøves, det finnes lite forskning på fenomenet og heller ikke standarder, validiteten er subjektiv, reliabiliteten snakker vi ikke om i kvalitative metoder, da det er intervjueren selv.

En vanlig måte å samle inn data på i kvantitative metoder er strukturerte spørreskjemaer med standardiserte spørsmål. *Fordeler* ved kvantitative undersøkelsesmetoder er at den tar fatt i det som er representativt og gjennomsnittlig for den gruppen som det forskes på. Dette fordi alle undersøkelsesenheter blir stilt ovenfor de samme spørsmålene i et spørreskjema. Det fører også til at man lettere kan generalisere og få en helhetlig oversikt, og se sammenhenger og mønstre som går igjen på det man forsker på. *Ulemper* ved det er at forskeren har liten fleksibilitet og

medvirkning i datainnsamlingen. Dette kan føre til at man overser viktig informasjon da spørreskjemaer kan bli rigide og kun gir svar på det forskeren selv mener er viktig.

3.3 Valg av metode

Vi valgte å gå videre med kvalitativ metode da vi så på den som mest relevant i forhold til vår problemstilling. Problemstillingen vår er en blanding av både teoretiske og beskrivende spørsmål, hvor vi sikter på å belyse, forstå og forklare handlinger og hendelser ved å ta utgangspunktet i et konkret prosjekt. Begrunnelsen for valg av kvalitativ metoden var vår tro på at den ville gi oss gode data og belyse spørsmålet vårt på en faglig interessant måte.

Kvantitativ tilnærmingen var ikke aktuell for oppgaven vår da undersøkelsesdesign og problemstillingen vår var eksplorativt, altså utforskende. Problemstillingen vår er hvordan man kan effektivisere prosjektstyringen og for dette kreves det at vi utforsker om hvordan RR definerer styring, hvordan det blir styrt i dag. Kvantitative undersøkelsesdesign er ment for problemstillinger som er mer av deskriptive, altså beskrivende og kausalt (årsaksforklaringer).

3.3.1 Intervju

Innenfor kvalitative metoder valgte vi å gå videre med intervju som datainnsamlingsmetode da personlige erfaringer, meninger eller lignende var av interesse. Intervjuene er med på å gi informantene en større frihet til å uttrykke seg . Dette mente vi ville gjøre det enklere for oss å få mer ut av spørsmålene som vi har skreddersydd.

Intervjuobjekter

Ut fra Scrum-teorien hadde vi på forhånd valgt å intervju de tre hovedrollene i Scrummetodikken:

- Scrum-master
- Utvikler (tre fra utviklingsteamet)
- Kunderepresentant

Disse fem intervjuobjektene ser vi på som mest hensiktsmessige å intervju, da det er mye kommunikasjon og samarbeid mellom disse partene i Scrum-metoden. Ved hjelp av disse fem hovedrollene vil vi få et bedre forståelse av hvordan de gjennomfører/styrer prosjektet fra tre forskjellige synsvinkler. Deretter analyserer vi dataene ved å tolke disse og finne ut om hvor forbedringspotensialene ligger. For å gjennomføre intervjuet lagde vi oss en intervjuguide og skreddersydde spørsmålene ut fra temaet og problemstillingen som vi har valgt.

Intervjuguide

Intervjuguiden ble utformet ved å ta hensyn til følgende punkter (Johannessen m.fl 2011)

- **Innledning**
Her presenterer vi hva hensikten med intervjuet er ved å spesifisere problemstillingen for oppgaven. Intervjuguiden vår er basert på flere hovedtemaer hvor hvert tema har en innledning som forklarer hva vi ønsker å finne ut ved det spesifikke temaet.
- **Faktaspørsmål**
Vi valgte å ha med faktaspørsmål for å bygge relasjon og tillitsforhold. Dette gjorde vi ved å stille generelle spørsmål om intervjuobjektet som navn, alder, stilling, erfaring og deres rolle i referanse prosjektet.
- **Introduksjonsspørsmål**
Her stiller vi spørsmål som er teoribaserte, for å få et bedre bildet av hvor godt intervjuobjektet kjenner til temaet fra før. Dette har vi gjort ved å stille intervjuobjektene spørsmål om hvordan de definerer ulike begreper innenfor temaet.
- **Nøkkelspørsmålene**
Nøkkelspørsmålene ble konstruert ut fra teorikapitlet. Her satte vi fokus på spørsmål vedrørende prosjektsyring og gjennomføringsmetodikken. Ved hjelp

av disse spørsmålene skulle vi få svar på det vi ønsket å utforske, slik at vi kunne bygge oppgaven videre på hovedproblemstillingen vår.

- Avslutning

Denne delen av intervjuguiden ble først og fremst utarbeidet for å oppklare eventuelle uklarheter under intervjuet og evaluere intervjuobjektets meninger om effektivisering av prosjektstyring. Det ble også fokusert på om det var noen kommentarer eller informasjon som intervjuobjektet ønsket å tilføye.

3.4 Datainnsamling

Det finnes tre måter å gjennomføre intervjuer på (Johannessen m.fl, 2011):

- Et *ustrukturert* intervju er uformelt, hvor spørsmålene tilpasser seg den enkelte intervjusituasjonen ved at spørsmålene og rekkefølgen på spørsmålene ikke er tilrettelagt på forhånd.
- Et *semistrukturert* intervju har en overordnet intervjuguide som utgangspunkt for intervjuet med temaer, spørsmål og rekkefølge. Men dette kan varieres ved at man beveger seg fram og tilbake i intervjuguiden.
- Et *strukturert* intervju gjennomføres ved fastlagte tema og spørsmål og det faste svaralternativer som forskeren krysser av for.

Intervjugjennomføringen var en blanding av semi- og strukturert struktur. Under gjennomføring av intervjuene tok vi utgangspunktet i intervjuguiden og prøvde å følge den kronologisk, men det viste seg at det var nødvendig med å gå fram og tilbake med temaer og spørsmål for å få mer utfyllende svar.

3.5 Dataanalyse

Kvalitativ dataanalyse innebærer å analysere data i tekstform. Å arbeide med tekst har bydd på spesielle utfordringer i analysen vår. Analyse av data i tekstform blir uttrykket som (Johannessen m fl., 2011):

Ord er feitere enn tall og kan tillegges flere meninger. Dette gjør dem vanskeligere å flytter rundt på og arbeide med. Enda verre, de fleste ord er meningsløse med mindre vi relaterer dem til ord som står foran og bak det ordet vi studerer (vår oversettelse).

I følge Johannessen m.fl har dataanalysen to hensikter:

- *Organisere data* som går ut på å redusere og systematiserer datamaterialet for å legge et godt grunnlag for analyse uten å miste viktig innhold.
- *Analysere og tolke* går ut på å tolke informasjonen fra flere perspektiver som ligger i datamaterialet.

Hvert intervju hadde en varighet på ca 45-60, min hvor det kom frem informasjon. Besvarelsene ble notert ned i punkter i et ark og i tillegg benyttet vi oss av elektronisk utstyr for lydopptak. Dette gjorde vi for å unngå å miste viktig innhold i intervjuet. Deretter analyserte vi innholdet ved å følge følgende analysestegene (Johannessen m.fl, 2011):

- Helhetsinntrykk og sammenfatning av meningsinnhold startet ved å gå igjennom notatene og renskrive lydopptakene for hver intervjuobjekt uten å fortape seg i detaljer. Deretter rangerte vi informasjonen i rekkefølge, da det var en del informasjon som kom frem under et tema som egentlig hørte til et annet.
- Koder, kategorier og begreper – det går ut på å finne meningsbærende elementer. Vi startet med å skille ut relevant stoff ved å markere viktige setninger i rødt.
- Kondensering går ut på å trekke ut alle meningsbærende elementer for å sitte igjen med redusert datamengde. Deretter kodet vi disse for å sette den relevante informasjonen sammen med informasjon med samme koding fra andre intervjuobjektene. Dette gjorde vi for å ha all relevant stoff innenfor en kategori et sted.
- Sammenfatningssteget går ut på å sammenligne sammenfatningsinntrykket med helhetsinntrykket som vi fikk i første fasen. Det viste seg at

sammenfatningsinntrykket samsvarte med helhetsinntrykket vårt. Vi fant ut til slutt at det fortsatt var noe tilleggsinformasjon som vi trengte for å styrke drøftingsgrunnlaget. Vi fikk muligheten til å intervju noen av utviklere på nytt og varighetene på disse var ikke mer enn 15 min. All data vi da fikk ble sortert ut umiddelbart.

3.5.1 Validitet og reliabilitet

Validitet dreier seg om gyldigheten og påliteligheten til de måleinstrumentene vi bruker. Altså tar det for seg hvorvidt en modell er presis og korrekt. Det finnes ingen måte å konstruere at en modell er korrekt eller ikke, men det finnes flere måter å antyde validitet:

Prediksjonsvaliditet – hvorvidt en modell klarer å predikere utfallet av en hendelse. Det er vanskelig å måle dette, men man kan i ettertid av en hendelse gå tilbake og se hvorvidt de faktiske resultatene korrelerte/samsvarte med de predikerte resultatene, og dømme modellene på bakgrunn av dette.

Innholdsvaliditet – hvorvidt denne modellen dekker det den faktisk skal dekke. Det er vanskelig å måle dette også, og det blir derfor subjektiv vurdering om hvorvidt modellen er valid med hensyn på innhold. Dette kan vurderes ut fra problemdefinisjonen.

Begrepsvaliditet – Det dreier seg om å teste sammenhengen mellom et teoretisk begrep og operasjonaliseringen av begrepet. Det er to forhold som er særdeles viktige her:

- Konvergent validitet – sier noe om hvorvidt indikatorer som antas å måle den samme teoretiske variabelen er høyt korrelert med hverandre). Hvis målingsinstrumentene korrelerer høyt med variablene er modellen konvergent valid.
- Diskriminant validitet – tester hvorvidt indikatorer som antas å måle ulike teoretiske begreper er lavt korrelert med hverandre. validitet sier noe om i hvilken grad et begrep skiller seg fra et annet begrep. Det operasjonelle mål

for begrep A må kunne skilles fra et operasjonelt mål av begrep B, og sammenblanding må unngås.

Kvalitative studier er ikke valide da de ikke kan kvantifiseres, altså måles (Johannessen m.fl, 2011). Vi har derfor evaluert validiteten ved å redegjøre hvilke metoder som er brukt for datainnsamlingen, intervjumetode, og analyse av transkripsjoner (hvordan innsamlingsdata er redusert og kategorisert).

Reliabilitet (pålitelighet) dreier seg om i hvilken utstrekning en måling vil gi det samme resultatet dersom det gjentas mange ganger. Alle målinger har tilfeldige feil, men en måling er mer reliabel dess mindre de tilfeldige feilene er. Man kan si at reliabilitet viser ”hvor godt vi måler det vi måler”.

Dette kan måles ved:

- *Test – retestmetoden*, hvor man gjennomfører undersøkelsen på et senere tidspunkt og sammenligner svarene fra begge gjennomføringen. Man får sjekket stabiliteten, men ikke hvorvidt andre variabler/faktorer gir samme resultater.
- *Intern konsistensmetoden*, Evnen til å produsere like resultater ved å benytte forskjellige utvalg til å måle et fenomen under den samme tidsperiode.

Evaluering av kvalitativ data med hensyn til reliabilitet er lite hensiktsmessig da man ikke benytter seg av strukturerte datainnsamlingsmetoder, observasjonen er verdiladet og kontekstavhengige og ikke minst fordi intervjueren i seg selv blir brukt som forskningsinstrument (Johannessen m.fl, 2011).

Kapittel 4 – Empirisk data

I dette kapitlet presenterer vi informasjonen som kom frem under intervjuene. Intervjuene startet med å stille generelle spørsmål for å bli bedre kjent med intervjuobjektene. Deretter satte vi fokuset på hovedtemaet prosjektstyring vi ønsket å få svar på. Ved avslutning av intervjuet stilte vi evalueringsspørsmål for å se om intervjuobjektene hadde noe mer å tilføye angående hva som kunne forbedres for å effektivisere prosjektstyringen.

Resultatet av intervjuene blir presentert ved hjelp av oppsettet som vi har utarbeidet i intervjuguidene (se vedlegg I, II og III). Vi starter først med å forklare hva prosjektet går ut på og avslutter med en modell som gir en oversikt over hvilke parter som er involvert i prosjektet. Deretter introduserer vi intervjuobjektene og fortsetter videre på følgende hovedtemaer:

- Fremdriftsplan
 - Tidsplanlegging
 - Ressursplanlegging

- Oppfølging
 - Kontroll
 - Kommunikasjon (Agile møter)
 - Oppdatering

Innholdet som fremkom under intervjuene er fremstilt på en nøytral måte. Resultatet vil deretter bli analysert og drøftet i neste kapittel. Vi har valgt å skille fremstillingen av intervjuene av ansatte i RR og kunden Flybuss Ekspress. Begrunnelsen er for å kunne belyse prosjektstyringen fra ulike synsvinkler.

4.1 Referanseprosjektet Mobile Ticket

Prosjektet går ut på å utvikle en total billettløsning for kunden Flybuss Ekspress AS.

Prosjektet er estimert til å skulle vare i 3 år og er delt i tre delprodukter:

- Mobilapplikasjon
- Webshop
- Billettmaskin

Mobilapplikasjonen er tidligere utviklet for kunden AKT (Agder Kollektiv Trafikk) og viste seg å være en interessant løsning for Flybuss Ekspress AS også. Ved hjelp av mobilapplikasjonen kan flybusskundene kjøpe periodebilletter via mobilene sine. I tillegg til mobilapplikasjonen var det et ønske om å ha et webshop hvor kundene kan kjøpe billetter via en webside også. Disse to produktene er per i dag på vei til å ferdigstilles.

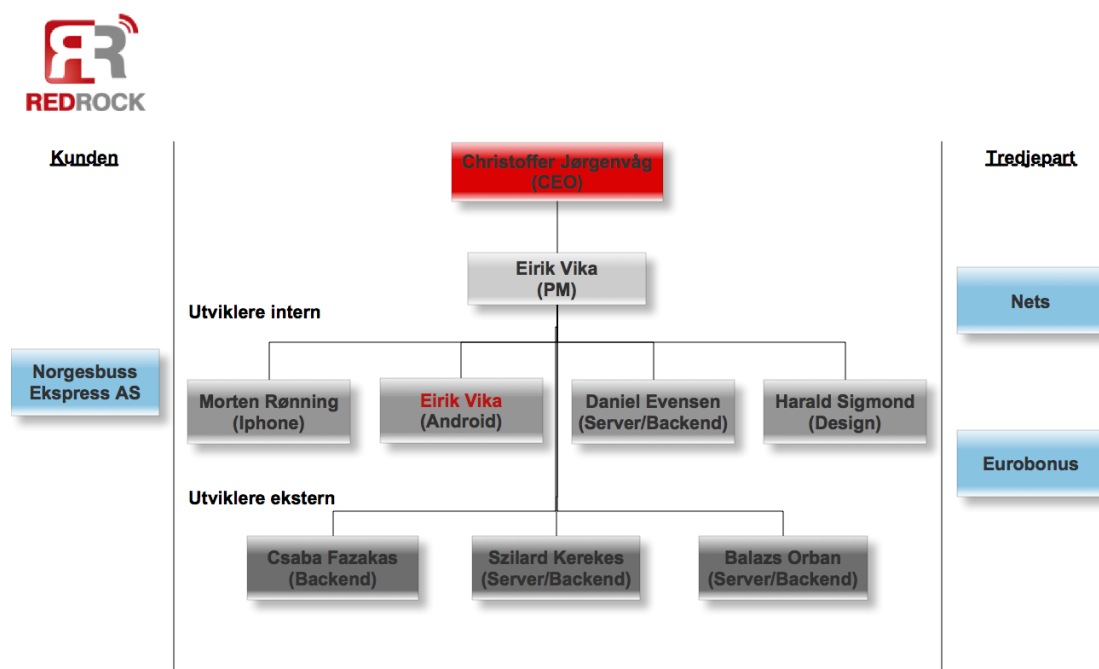
RR har nylig satt i gang med utvikling av det tredje delproduktet, billettmaskinen. Billettmaskinen skal ferdigstilles som et fysisk betalingsautomat, som bussjåførene skal ta i bruk på bussene. Alle disse delproduktene skal være integrerte med hverandre i henhold til de kravene og spesifikasjonene Flybuss Ekspress AS og RR AS har definert. Vi valgte å rette intervjuene mot delproduktene mobilapplikasjonen og webshop. Dette fordi det foreligger mer informasjon om prosjektstyring under utvikling av de to første delproduktene enn delproduktet, billettmaskinen, som de nettopp har startet med.

Det er også involvering av to stk tredjeparter i dette prosjektet:

- Nets
- Eurobonus

Nets leverer betalings-, kort- og informasjonsløsninger i de nordiske landene og spiller en stor rolle i fullførelse av prosjektet da produktene skal integreres med denne betalingsløsningen. Eurobonus er et bonusprogram som tilbyr bonusreiser, service fordeler og tilbud til sine samarbeidspartneres kunder.

Under utvikling av delproduktene kreves det at både RR og Flybuss Ekspress har god kommunikasjon med tredjepartene da produktene skal integreres med både betalingsløsningen og bonusprogrammet. Nedenfor har vi laget et organisasjonskart for å gi en bedre oversikt av hvilke parter som er involvert i prosjektet Mobile Ticket.



Figur 21– Organisasjonskart for prosjektet Mobile Ticket

4.2 Intervju med Redrock

Referanseprosjektet Mobile Ticket består som tidligere presentert av tre delprodukter. Første og andre delproduktet er allerede levert og RR jobber nå med det tredje delproduktet. Det har vært to prosjektledere for dette prosjektet, en for delproduktene en og to, og en for delproduktet tre. Vi startet intervjurunden med prosjektlederen for delproduktet 3, da han også jobbet som utvikler for delproduktet en og to.

Intervjurunden startet gjort fordi han hadde innsikt i prosjektstyringen fra både utviklers og prosjektleders side, og vi kunne få et fundamentet på det overordnede nivået av styringen. Deretter intervjuet vi tre utviklere, to fra RR i Grimstad og en fra RRT, og kunderepresentanten og prosjektlederen som ledet del 1 og 2 av prosjektet Mobile Ticket. Utvikleren fra RRT befant seg i Romania og kunderepresentanten befant seg i Oslo på intervjutidspunktet, derfor ble disse intervjuene utført via

kommunikasjonsprogrammet Skype. Alle intervjuene ble tatt opp med opptak for mer presis etterarbeid. Prosjektlederen for del 1 og 2 av prosjektet hadde dessverre ikke tid til å ta imot oss for et direkte intervju. Det ble derfor utført ved at vi sendte intervjuguiden til han på e-post, og han besvarte våres spørsmål skriftlig. Innholdet som fremkom under intervjuene blir fremstilt på en nøytral måte nedenfor. Resultatet vil deretter bli analysert og drøftet i neste kapittel.

Fremdriftsplan

Fremdriftsplanen for prosjektet var bygd opp basert på endringer i funksjonalitet i forhold til lignende mobilbillettprosjekt, og ny funksjonalitet basert på kundens ønsker. Planen ble utarbeidet av prosjektlederen i samarbeid med en utvikler . Den estimerte varigheten på prosjektet var tre måneder. Ved spørsmål om hvilket grunnlag RR beregnet dette på var at de ” tok utgangspunkt i tidligere mobilbillettutvikling og vag spesifisering”.

Delproduktene mobilapplikasjon og webshop bestod av følgende milepæler:

- Serverapplikasjon for mobilnettet
- Innloggingsside for brukere hvor de kan registrere et kort for betaling
- Webshop for kjøp av billetter
- Backoffice kontrollpanel - det kunden kan bruke for å administrere løsningen
- Iphone app for billetter
- Android app for billetter

Milepælene ble videre delt inn i sprinter, som igjen bestod av oppgaver, hvor det da ble planlagt detaljert med utviklingsteamet. En sprint hadde en varighet på 14 dager. Forslaget for fremdriftsplanen ble først sendt til kunden for tilbakemeldinger før RR satte i gang med utførelse av prosjektet.

Ved utbygging av fremdriftsplanen ble det også satt inn noen faktorer som RR kaller for ”dependencies” som er en kritiske faktorer for at produktet skal fungere.

Feilestimeringer av milepælene førte til at RR endte opp med flere

replanleggingsprosesser og ikke minst forsinkelser i prosjektet. Dette kunne vært unngått dersom flere utviklere var med i estimeringsprosessen av tid og kompleksitet, hvor de også hadde fått en bedre forankring i prosjektet. ” Fremdriftsplanen kunne vært forbedret med en mer detaljert spesifisering og kunnskapsoverføring ” i følge prosjektleder. RR ønsker videre å effektivisere styringen ved å ha bedre kontroll og oversikt over hvilke ressurser som er tilgjengelige og ikke minst involvering av og forpliktelse fra linjeledelsen.

Rolle og oppgaver

Under intervjuet kom det frem at rollene i Scrummetodikken ikke ble fulgt helt etter Scrumteorien. I tillegg til rollen som prosjektleder, hadde prosjektlederen også oppgave som utvikler. Prosjektlederen var også involvert i andre prosjekter med begge rollene sine.

Ved spørsmål om hvordan han håndterte det svarte han:

” Det ble vanskelig å holde fokus på bare prosjektledelse da jeg selv hadde egne tidsfrister på Androidutviklingen. En prosjektleder burde ikke utvikle noe, men ha fokus på å lede prosjekter.”

Opp til dette prosjektet hadde han ikke arbeidet som prosjektleder i et utviklingsprosjekt som var i like stort omfang som dette.

Tidsplanleggingen

RR benytter seg av prosjektstyringssystemet Jira som er en del av Atlassian Tools. Jira ble brukt til tidsplanlegging-, timeføring- og oppretting av arbeidsoppgaver i en sprint.

Oppgavene ble utarbeidet ut fra funksjoner som ble hentet inn fra produktbackloggen hvor funksjonene var rangert i prioritert rekkefølge. Antallet på funksjoner var avhengig av hvor mange funksjoner som kunne tilpasses i en sprint, altså 14 dager.

Det ble utarbeidet en sprint om gangen hvor x antall funksjon(er) ble detaljert planlagt. Det ble også utarbeidet user stories for komplekse funksjoner for å få en bedre oversikt over hvilke bindinger hver oppgave hadde i forhold til hverandre i riktig rekkefølge.

Alle oppgavene var rangert i prioritert rekkefølge. Hver oppgave ble videre detaljert planlagt av prosjektlederen sammen med utvikleren som skulle utføre den utvalgte oppgaven. Dette startet ved at oppgavene ble rangert ut fra deres kompleksitet:

- Blocker
- Critical
- Major
- Minor
- Trivial

Om en oppgave var rangert som ”blocker”, betydde det at oppgaven hindret andre oppgaver i sprinten med å starte. Dermed måtte ”blocker-oppgaven” ferdigstilles før andre oppgavene kunne påbegynnes. Disse ble sett på som startaktiviteter. Andre utviklerne kunne imidlertid starte med andre oppgaver som ikke var blokkerte, altså oppgaver som var uavhengige andre oppgaver. Varighetene på oppgavene ble estimert ut ifra deres kompleksitetsgrad (critical, major, minor og trivial) og utviklernes erfaring med liknende oppgaver.

RR hadde tidligere benyttet seg av estimeringsmetoden planning poker for å estimere tid og kompleksiteten, men valgte å slutte med det da det tok for lang tid. Det ble derfor avgjort om å estimere oppgavenes varighet med en kort diskusjon mellom prosjektlederen og utvikleren. RR har en rutine på at en oppgave ikke skal estimeres over 7,5 timer, altså en vanlig arbeidsdag. Ut i fra dette ble det tildelt minimum 10 oppgaver per utvikler per sprint. Oppgavene ble delt inn i deloppgaver dersom det var noen oppgaver som krevde mer enn 7,5 timer.

Verktøyet burndown chart i Jira ble brukt av prosjektlederen og utviklerne for å holde styr på hvordan fremdriften var i forhold til den budsjetterte planen. Utviklerne førte inn antall timer de hadde brukt på oppgaven og deretter beregnet systemet automatisk ut hvor mange timer det var igjen ut ifra den estimerte tiden. Etter hvert ble sprintene

unøyaktige av endringer og nye oppgaver som hastet, så burndown-charten ble ikke lenger tatt i bruk.

Utviklerne testet sitt eget arbeid etter fullførelsen av alle oppgavene i en sprint og deretter ble dette dobbeltsjekk av prosjektlederen før det ble demonstrert for kunden. Dersom det var noen feil, ”bugs”, måtte dette fikses med en gang. Hvis noen av oppgavene i en sprint ikke ble ferdige til fristen, altså 2 uker, ble de flyttet videre til neste sprint. Og hvis det viste seg at alle oppgavene i en sprint ble ferdige før fristen ble oppgavene fra neste sprint hentet inn.

Kostnadsplanlegging

Kostnadsplanleggingen ble som oftest utarbeidet av bedriftens daglig leder. Estimeringen RR gjør baserer seg på deres internkostnader og hva estimerte timer i prosjektet er. Kostnadene for ressursene estimeres ut i fra internkostnaden og ved å se på fremdriftsplanen. De største kostnadspostene RR hadde for prosjektet Mobile Ticket var timeforbruket på grunn av dårlig planlegging og ” Lite detaljert spesifikasjon og muntlige avtaler som gjorde en kunde krevende på diverse punkter.” Kostnadene for prosjektet var basert på timeforbruket. Ved kostnadsoppfølging benyttet RR Jira som prosjektstyringsverktøy som ga dem oversikt over timeforbruket i forhold til det som hadde blitt estimert for oppgavene. RR hadde en avtale om fastpris med kunden. I Jira fantes det verktøy for oppfølging av timeforbruket, som for eksempel rapport for hvor mye timer som er forbrukt i løpet av en sprint, men det var RR dårlige med å benytte seg av.

Ressursplanlegging

Utviklerne i prosjektet ble sett på som ressurser under intervjuet. Det kom frem at RR ikke hadde noen system for allokering av ressurser som gjorde at de ikke hadde en god oversikt over hvilke ressurser som var tilgjengelig til hvilken tid. Prosjektlederen skulle sørge for at en utvikler ikke satt med oppgaver fra to forskjellige prosjekter på en dag og prøvde å legge opp dagsoppgavene slik at de kunne fokusere på et prosjekt om gangen, selv om det ikke alltid gikk.

Utviklingsteamet bestod av utviklere med ulike kompetanser på ulike nivåer. Det bestod blant annet av en *server ansvarlig* som jobbet med API'er (kommunikasjonen mellom server og klienter), en *android ansvarlig* og *iphone ansvarlig* da produktet skulle tilpasses begge operativsystemet på mobiltelefonene og en *design ansvarlig* som hadde ansvaret for layouten av mobilapplikasjonen. Design ansvarlig var den eneste i utviklingsteamet som kunne håndtere designoppgavene. I dette prosjektet ble designeren og serveransvarlig sett på som mest kritisk, da det ikke var noen andre i teamet med lignende kompetanse innenfor design og serveransvarlige hadde erfaring fra et tidligere mobilbillettprosjekt.

I tillegg til utviklerne ble nødvendig informasjon fra kunden og kommunikasjon med tredjeparts leverandør sett på som kritiske ressurser for å oppnå milepælene. Disse oppgavene ble da tiltenkt til å tas så tidlig som mulig. Tredjepart i dette prosjektet var betalingsleverandøren hvor kommunikasjon med dem under integrering av mobilapplikasjonen var nødvendig for at produktet skulle fungere optimalt. I tilfeller hvor RR ikke var sikre på hvor lang tid noen oppgaver skulle ta på grunn av mangel på ressursene, doblet de tidsestimatene for å være på den sikre siden.

Oppfølging

I oppfølgingen av dette prosjektet ble det holdt tre ulike møter; daglige status (Scrum) møter, ukentlige møter og sprint møter. Tidligere har vi nevnt at Jira ble tatt i bruk for å se hvor utviklerne lå i forhold til oppgavene de skulle ha utført i henhold til planen. Jira kunne blitt brukt for oppfølging av oppgavens og prosjektets framgang ved hjelp av burndown chart, men sprintene ble unøyaktige av endringer og nye oppgaver som hastet. Prosjektlederen brukte Jira som utgangspunkt for de daglige statusmøtene med utviklerne. I følge prosjektlederen var møtene effektive i forhold fortløpende oppdatering av statuser og til oppfølging av utviklerne som hadde mer behov for motivering og kontroll. Siden alt skjedde ved muntlig kommunikasjon ble skriftlige rapporter og dokumentasjon lite brukt. RR mener skriftlig dokumentasjon er noe de kan bli bedre til. Kun backloggene ble oppdatert ved behov. Utviklerne oppdaterte sprintbackloggen hver dag om hvor langt de hadde kommet med oppgavene de skulle ha gjort, hva som ble testet og hva som feilet. Var det store ting som feilet ble det tatt

opp omgående. Var det noe som kunne ventes med ble det tatt opp på neste møte. Da de daglige møtene var en-til-en kommunikasjon ble det også holdt ukentlige møter for statusoppdatering med hele teamet samlet. Produkt backloggen ble da oppdatert om det var noe spesielle endringer i funksjoner som skulle leveres kunden. Det fremkom av prosjektlederen at ”I dette prosjektet ble det mye endringer i backloggen grunnet lite detaljert spesifisering og ikke alle oppgaver ble tenkt i starten”, men det var sjeldent at rekkefølgen av oppgavene ble omprioritert.

Sprintmøtene ble holdt annenhver uke for blant annet å oppsummere hva som har blitt gjort i sprintperioden og for å planlegge neste sprint. Disse møtene kunne finne sted med bare en av utviklerne fra teamet, ikke alltid alle kunne delta. Sprintbackloggen ble deretter oppdatert fortløpende. Dersom det kom frem noen feil i funksjonene ved testing ble dette lagt inn i sprintbackloggen med en gang. Feilene som var kritisk ved funksjonene ble tatt hånd om øyeblikkelig mens de andre småfeilene kunne ventes.

Når det kom til rapportering til eksterne parter som for eksempel kunden RR rapporterte fremgangen til kunden ved å levere timelistene annen hver uke slik at kunden hadde oversikt over hva de betalte for. Statusrapport over gjenstående oppgaver for at prosjektet skulle bli godkjent ble og opprettet, som prosjektleder kommuniserte med kunde om.

De typiske årsakene til at det ble hindringer eller forsinkelser i prosjektet var ofte fordi spesifisasjonene av funksjonene var for dårlig spesifisert i tilbudet som ble gitt til kunden. Spesifisasjonene var ofte for vage som førte til at kunden tolket det annerledes enn det RR hadde sett for seg. Situasjoner som dette førte til diskusjoner og/eller uenigheter under gjennomføring av prosjektet hvor konsekvensene var forsinkelse av prosjektet. I forhold til metodikken Scrum hvor endringer fra kunden ønskes velkommen for å kunne gi en effektiv løpende kundetilpasning avviker gjennomføringen til RR fra det.

Feilestimeringer kunne være en annen årsak til forsinkelse i prosjektet. Alternative løsninger kunne være å ta vekk noen funksjoner, men det avhenger av kunden. Noen kunder er mer krevende enn andre. RR måtte da ha en god begrunnelse på hvorfor det ikke er mulig, og det kunne være ting som de ikke kunne forutse fra starten av. Kunne

hende at noen funksjoner som RR hadde i tilbudet ikke var mulig i realiteten. Slike ting kan skje med for eksempel iPhone fra Apple, hvor det er veldig begrenset på løsninger som kan finnes.

Dersom det oppstod problemer eller hindringer, diskuterte prosjektlederen problemet med utvikleren det gjaldt og prøver å finne fram en løsning.

Om prosjektleder og utvikler ikke kunne løse problemet innkalte prosjektleder gjerne flere utviklere for å kunne se på problemet med nye øyne. Måtte oppgavene forskyves, ble de flyttet fra den ene sprinten til den andre. Dersom den andre sprinten blir overbelastet må noen av oppgavene flyttes tilbake til produktbackloggen.

I slike situasjoner ble løsningen ofte en eller to ekstra sprinter som da førte til forsinkelse av prosjektet. RR løser forsinkelser oftest ved å jobbe overtid.

Betalingsleverandøren har vært det største problemet hittil med tanke på forsinkelse grunnet en ekstern part som sitter med informasjon hvor RR ikke har kontroll over.

Evaluering

Både utviklere og prosjektleder syntes at kommunikasjon mellom teamet og med kunde har vært bra.

Med tanke på hvordan prosjektstyringen kunne vært effektivisert var det gjennomgående blant hele teamet at det kunne vært mer fokus på selve prosjektstyringen. Det at prosjektleder hadde to roller gjorde det vanskelig å allokere tid til alt. Oppgavene kunne vært klarere definert, hvor man skulle gått dypere inn i kompleksiteten og spesifikasjonene. Det kunne ha blitt gitt et mer helhetlig bilde av hele prosjektet, noe en av utviklerne også savnet. Utviklerne savnet å kunne lett få tilgang til hvor i fremdriftsprosessen man er i prosjektet og ha et overordnet informasjonsbilde.

RR syntes at de har vært litt for fleksible ovenfor kunden. Det ble gjort en god del muntlige avtaler hvor kunden kunne komme med endringer når som helst og det ble

akseptert av prosjektleder. Det kunne komme av at spesifikasjonene var vage mente teamet.

I følge prosjektleder var årsakene til forsinkelsen at det var for lite detaljert spesifikasjon, mange muntlige samtaler om oppgaver som skulle gjøres, lite kunnskapsoverføring fra tidligere mobilbillettprosjekt og dårlig prosjektledelse. Til ettertanke fremkom det at forsinkelsen kanskje kunne vært forhindre om prosjektet hadde mer detaljert spesifikasjon, som da ikke ville gitt ”muntlige avtaler” om diverse funksjonalitet og oppgaver. Andre faktor som kunne ha bidratt til å forhindre forsinkelsen var at man kunne fått bedre kunnskapsoverføring fra tidligere prosjekt, teamet kunne ha fulgt agilt prosjektledelse bedre, og de kunne vært nøyere med estimeringen.

4.3 Intervju med Flybuss Ekspress AS

Intervju runden med kunden ble også gjort via kommunikasjonsprogrammet Skype da kunden befant seg i Oslo. Vi valgte å intervju kunden til sist siden vi satt med mer kunnskap om hva slags involvering kunden hadde i prosjektstyringen og kunne gjøre endringer i intervjuguiden ved behov.

Prosjektet

Intervjuobjektet som driftsutvikler for Norway Bussekspress. Intervjuobjektet er en av fire som var med i prosjektgruppen hos Norway Bussekspress, som skulle følge opp samarbeidet med RR. Han hadde IT-utviklingskunnskap fra tidligere arbeid, men hadde ikke kjennskap til teorien om agile arbeidsmetoder. Ingen i gruppen hadde det. Samarbeidet med RR gikk ut på prosjektet til mobilapplikasjonen for Flybuss ekspressen fra Oslo til Gardemoen som er eid av Norway Bussekspress AS. Tommys og gruppens oppgaver var å støtte RR i prosjektet og se på den tekniske delen da de hadde kjennskap til det fra sjåførens side. Målet var at kunden skal kunne kjøpe bussbillett via applikasjonen på mobilen sin for Flybussen ekspressen.

Fremdriftsplan

Intervjuobjektet mente at fremdriftsplanen var ”i overkant ambisiøs” til å begynne med. Spesifikasjonene var gode nok, men hans syntes tidslinja som var litt stram. Norway Bussekspress (forkortes herved til NB) ville selvfølgelig ha produktet levert så fort som mulig, men var villig til å bruke 2-3 kvartaler på prosjektet. De hadde nemlig opplevd at samme prosjekt med et annet selskap ikke ble en suksess tidligere og ville bruke tid til å få det til denne gangen. Likevel ble prosjektet med RR levert forsinket. ” Bedre å levere sent og godt, enn tidlig og dårlig” var Tommys ord for forsinkelsen. Det viste seg at det først og fremst var kvaliteten på produktet som ble prioritert. Når det kom til planlegging fikk Gruppen fra NB ytret sine meninger om hva produktet skulle inneholde. NB har vel ønsket mer enn det som var avtalt i fremdriftsplanen, og det har endt opp med at noen funksjoner måtte tas bort på grunn av ikke alt var mulig å få til. Når vi spurte om Tommy hadde kjennskap til Scrum, svarte han at det hadde han ikke. Vi stilte spørsmål om produkt backloggen:

”Var du involvert i utarbeidelsen av produkt backloggen? I så fall deltok du i rangeringen av oppgavene?”

” Jeg deltok i å bestemme hva som skulle leveres først. Det ble satt en spekk i fellesskap på hva produktet skulle inneholde og RR skulle sette en fremdriftsplan på når de forskjellige tingene skulle være på plass. Vi godkjente planen, med den ambisiøse fremdriften som lå der.”

Videre spurte vi om de kunne tenkt seg å legge av en dag til å planlegge produkt backloggen, men de mener at det er vanskelig å legge av ressurser til det.

NB fikk en plan som var på et overordnet nivå, hvor det ble satt et spekk på design. Når vi spurte om de fikk se noen milepælplan var svaret at det kanskje ikke var noe milepælplan, de følte at fremdriftsplan og milepælplan lå i samme overordnet plan. NB fikk se den planen en gang, og ellers fikk de ikke høre om noen oppdateringer. Angående timeforbruket hadde de ingen oversikt over, det var et ønske i etterkant at de kunne fått en god oversikt over hva det var blitt brukt tid på.

Til uttesting av mobilapplikasjonen fikk de en liste utlevert. Dersom det var ting som var galt, sa Norges busser i fra og feilen ble fikset som regel bare på noen timer.

Ressursplanleggingen hadde ikke NB satt seg inn i. Der lot de RR styre. Tommy visste at det var en utvikler til hver interface, men ellers visste han ikke så mye om hvordan RR planla ressursene sine.

Nets og Eurobonus ble angitt som kritisk ressurser. RR kommuniserte direkte med betalingsleverandøren Nets, da de hadde erfaring med dette. NB var dermed ikke bekymret angående denne kommunikasjonen. Bonusavtaler via Eurobonus anså han ikke som like kritisk for det kunne alltid ordnes etter hvert da all transaksjonsdata ville være lagret. Det er altså poeng som registreres på profilen til kunden basert på billettsalget, og salget kan alltid etterregistreres.

Oppfølging

Intervjuobjektet oppga at han ikke kjente til Scrummetodikken, intervjueren fant da ut at det ikke var poeng i å bruke teoretiske ord fra metoden for spørsmålene og intervjuet videre basert på generelle oppfølgingsspørsmål.

Til oppfølgingsdelen av intervjuet av intervjuet kom det frem at kunden ikke hadde faste møter med utviklingsteamet, men sporadiske etter behov, det vil si når RR hadde noe å levere. De har nå lært fra det forrige prosjektet at de må se hvordan produktet ser ut tidligere, og kjører nå ukentlige statusmøter for å følge opp prosjektet.

Evaluering

Til evaluering hadde vi noen oppsummerende spørsmål til kunden for å få kunnskap om hva han syntes om prosjektgjennomføringen og -styringen. Kunden mener at han har egentlig tilført prosjektet all nødvendig informasjon til utviklingsteamet for utarbeidelsen. Han var videre veldig fornøyd med kommunikasjonen med RR, ”de responderte raskt og leverte fort.” I prosjektstyringen savnet Tommy et rammeverk for prosjektet. Han var vant til tunge dokumentasjoner. RR kunne vært dyktigere på å strukturere all dokumentasjonen. Noe kunden ønsket mer av var ukentlig

statusoppdatering på hva utviklingsteamet har utført. Fra kundens side kunne det også ha vært en mye klarere deltakelse for å gi en ekstra drivkraft.

Kapittel 5 - Analyse

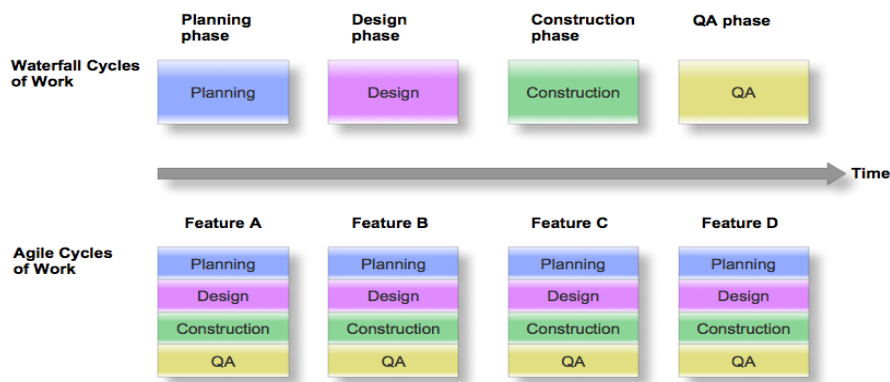
I dette kapitlet starter vi med å analysere styringsmodellene; tradisjonelle og agile ved å sammenligne de. Deretter analyserer vi informasjonen som kom frem under intervjuene og undersøkelsene for referanseprosjektet til RR. Referanseprosjektet analyseres ved å ta utgangspunktet i agile metoden Scrum i samsvar med sammenligningen av styringsmodellene. Til slutt drøfter vi forbedringspotensialet ved å identifisere faktorer som vi mener ligger til grunn for svikt i effektiv prosjektstyring og foreslår tiltak til dette.

5.1 Tradisjonelle VS Agile

Tradisjonelle- og agile metoder baserer seg på to forskjellige prosjektstyringsmodeller som gjør at prosjektene blir styrt og gjennomført på forskjellige måter. Tradisjonelle metoder har sin opprinnelse fra bygg- og anleggsindustrien (Gottschalk og Gottschalk, 2012). Agile metoder har derimot sin opprinnelse fra programutviklingsbransjen da iterative designmodell/tanke er nødvendig i design-, data-, forskningsprosjekter (noen).

5.1.1 Forskjeller

Hovedforskjellen mellom tradisjonelle- og agile metoder er at tradisjonelle metoder er lineære mens agile metoder er ikke-lineære. Agile metoder følger med andre ord en iterativ strategi istedenfor en lineær strategi og har fokus på en adaptiv prosjektstyring (Fernandez og Fernandez, 2008). Den iterative modellen består av samme prosjektfaser som det er i tradisjonelle metoder, men skiller seg ved å dele prosjektet inn i flere sykluser. Figur 22 viser faseinndelingen i både tradisjonelle- og agile metoder.



Figur 22 – Sykluser av arbeid i tradisjonelle - og agile metoder (fritt etter Layton, 2012)

Sekvensielle gjennomføringer egner seg i for eksempel gjennomføringsfasen i et fysisk tilvirkningsprosjekt, som et byggeprosjekt. Man kan for eksempel ikke prøve og feile i selve byggefasen, da dette ville blitt ekstremt dyrt. Iterative gjennomføringer egner seg ikke for et fysisk tilvirkningsprosjekt, men heller for IT- prosjekter. Produktkravene blir delt opp i *features* og følger fossefallmodellens faser ved gjennomføring av de, slik vi ser i figur 22.

I tradisjonelle metoder er kundekravene veldokumentert og forankret i planleggingsfasen, der krav og behov avstemmes mot design. I agile prosjekter oppdages komplette prosjektkrav ved å gjennomføre iterasjoner og er derfor med på å eliminere usikkerheten. De følger strategien ”learn-by-doing”. Dette fører til at agile metoder har høyere risiko i forhold til tradisjonelle prosjekter, men agile metoder har fleksibiliteten til å lettere justere endringer i prosjektkravene (Fernandez og Fernandez, 2008).

Tradisjonelle prosjekter styres ved å ta hensyn til budsjett, tidsplan og omfang i lik nivå (Fernandez og Fernandez, 2009). Beregninger og avvik kan derfor lettere bli sporet opp i tradisjonelle prosjekter, da man ønsker å redusere risikoen ved å ta hensyn til begrensninger av tid og penger. I motsetning, går agil prosjektstyring ut på å fokusere på leveranser og forretningsverdi. Budsjett og tidsverdi er sekundære for agil prosjektstyring (Fernandez 2009).

5.1.2 Likheter

Ut fra teorien vi har redegjort under delkapitlene for tradisjonelle- og agile metoder ser vi at det foreligger noen likheter. Likhetene har forskjellige navn, men fremgangsmåten er det samme.

Prosjektfaser

I teori blir Agile metoder sett på som en moderne måte å styre prosjekter på, men i praksis følger man tradisjonelle metoden ved å dele produktet (prosjektet) i flere delprodukter som gjennomføres i de samme prosjektfasene slik det fremkommer i en fossefallsmodell (se figur 5). Layton (2012) forklarer denne likheten ved:

Agile methodology means cutting down the big picture into puzzle size bits, fitting them together when the time is right e.g. design, coding and testing bits.

Prosjektnedbrytning

I tabell 9 har vi utarbeidet prosjektnedbrytningsstruktur for agile prosjekter ved å ta hensyn til tradisjonell nedbrytningsstruktur. Ved hjelp av tabellen kan vi se hvordan milepælplan kan sammenlignes med en releaseplan. Dette eksemplifiseres med en WBS- oppdeling.

Nivå	Hensikt	Tidshorisont	Level	Purpose(e) (Examples from table 3 and 4)	Duration
Prosjekt	Mål	9-18 måneder	Project	Goal (Mobile bank application)	Depends on number and duration of releaseplans
Delprosjekt	Delresultat	6-12 måneder	Theme	Releaseplan (See account balances)	Depends on number and duration of sprints
Arbeidspakke	Milepæl	1-3 måneder	Features	Sprinter (see checking account balance)	(1-4 weeks)
Aktivitet	Målbart resultat	1-3 uker	Epic User Stories	Measureable result (see checking account balance)	Depends on number and duration user stories
Arbeidsoppgave	En større arbeidsinnsats	Dager	User Stories	A greater workload (Log into mbolie account)	Days
Jobb	Kortvarig arbeidsinnsats	Timer	Task	Short-term workload (Create an authentication screen for a username and password, with a Submit button)	Hours

Tabell 9 – Sammenligning av WBS (fritt etter Karlsen, 2012)

En milepælplan består av milepæler som er en beskrivelse av en tilstand. Dette kan for eksempel være en beskrivelse av en leveranse som prosjektet bør ha kommet til i et visst stadium av prosjektet. Det er nettopp det en releaseplan har som formål. Iterasjoner (sprinter) i en releaseplan kan ses på som milepæler i en milepælplan. Det er nettopp denne planen utviklingsteamet forholder seg til og leverer til kunden før de starter med neste releaseplan. Ut fra prosjektets størrelse og krav om antall leveranser i prosjektet, kan et prosjekt inneholde flere releaseplaner (milepælplaner).

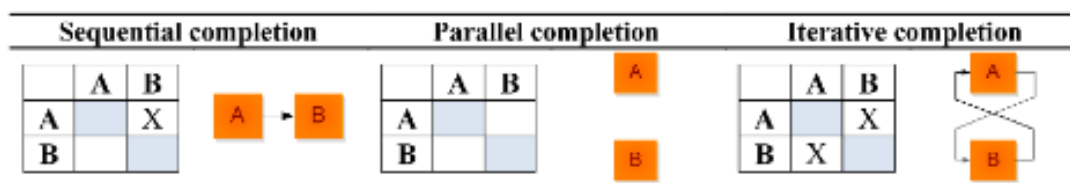
5.1.3 Fremdriftsstyring

Fremdriftsstyring i tradisjonelle metoder baserer seg på nettverkplanleggings metoder som CPM, PERT og Gantt. Det hevdes at disse metodene er på det operasjonelle nivået – logisk planlegging, tidsestimering, kostnad og ressurs fordeling, mens agile metoder krever metoder på det strategiske nivået (Zsolt, 2012). Det operative nivå tar sikte på den daglige driften, mens det strategiske nivået har mer langsiktig fokus under planlegging (Jacobsen og Thorsvik, 2005). I følge teorien under agile metoder hevdes det at man planlegger jobber mot en releaseplan om gangen. Denne planen blir deretter delt i sprinter til arbeidsoppgaver. Denne planen baseres jo da på det operativ nivået, da man har daglige oppgaver, oppfølginger, endringer osv. Fremdriften i agile metoder er mer dags basert enn lagsiktig og vi vil derfor konkludere med at agile metoder krever mer av operativ enn strategiskstyring. Strategisk styring er i lik med operativ styring like nødvendig, og gjelder for alle selskaper uavhengig styringsmodell man velger.

Mohammad (2013) hevder at aktivitetsvarighetene i CPM utarbeides ut fra hvilke ressurser som trengs for å fullføre en aktivitet, og ikke ut ifra hva som er tilgjengelig. Etter vår forståelse av teorien så mener vi at det blir tatt hensyn til tilgjengelig ressurser i CPM, ved for eksempel replanlegging. Hensikten med oppfølgingen er å oppdatere planen for å se om hvordan man ligger an i forhold budsjettert plan og dersom det viser seg at det er avvik så vurderes det tiltak for det. For å sette i gang tiltakene kreves det at man replanlegger og da gir det ingen mening med å replanlegge hvis man ikke har tatt hensyn til tilgjengelig ressurser. Vi vil dermed konkludere med

at man i CPM tar hensyn til tilgjengelig ressurser når man utarbeider en fremdriftsplan.

I følge Zsolt tar matrix baserte metoden hensyn til tre avhengighetsforholdene under tidsplanleggingen 1) sekvensiell gjennomføring, parallell gjennomføring og 3) iterativ gjennomføring. Sekvensiell avhengighetsforholdet vil si at B kan starte etter at A er ferdig, parallell forholdet vil si at oppgavene skal utføres parallelt og iterativ vil si at etter at B er ferdig, kan A gjennomføres igjen. Zsolt (2012) hevder at man kan finne avhengighetsforholdene 1) og 2) i CPM og PERT, men ikke den iterative forholdet som er hovedforholdet i iterative prosjekter. Dersom en aktivitet blir delt i to eller flere mindre aktiviteter kan iterative forhold oppdages. Dersom det blir gjort noen endringer i en mindre aktivitet så kan det føre til at en annen mindre aktivitet må utføres igjen for å fullføre hovedaktiviteten. Vi ønsker å underbygge påstanden ved at avhengighetsforholdet iterativ befinner seg også i CPM.



Figur 23 – Avhengighetsforholdene mellom aktivitetene (Zsolt, 2011)

Tabell 10 oppsummerer hovedforskjellene mellom tradisjonelle - og agile prosjektstyringsmodellene.

Time Management with Traditional Approaches	Time Management with Agile Approaches
Fixed scope directly drives the schedule.	Scope is not fixed on agile projects. Time can be fixed, and development teams can create the requirements that will fit into a specific time frame
Project managers determine time based on the requirements gathered at the beginning of the project.	During the project, scrum teams assess and reassess how much work they can complete in a given time frame.
Teams work on all project requirements at one time in phases, like requirements gathering, design, development, testing, and deployment. There is no schedule difference between critical requirements and optional requirements.	Scrum teams work in sprints and complete all the work on the highest-priority, highest-value requirements first.
Teams do not start actual product development until later in the project, after the requirements-gathering and design phases are complete.	Scrum teams start product development in the very first sprint.
Time is more variable on traditional projects.	Time-boxed sprints on agile projects stay stable.
Project managers try to predict schedules at the project start, when they know little about the product.	Scrum teams determine long-range schedules on actual development performance in sprints. Scrum teams adjust time estimates throughout the project as they learn more about the product and the development team's speed, or velocity. You find more about velocity later in this chapter

Tabell 10 – Tradisjonell VS Agil (fritt etter Layton, 2012)

5.2 Resultater og drøfting

Hovedproblemstillingen for denne oppgaven er

”Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?”

RR har i den siste tiden opplevd en del forsinkelser på sine prosjekter og håper at forbedret prosjektstyring kan forhindre dette. De ønsker med andre ord en bedre kontroll over fremdriftsplan og oppfølging av prosjektene sine. I dette kapitlet vil vi drøfte årsaker som kan ha forårsaket ineffektiv prosjektstyring som blant annet har ført til forsinkelser.

5.2.1 Styringsmodell

Ved gjennomgang av informasjonen fra intervjuene fremkommer det at RR har tilegnet seg Scrumtilnærmingen ved å tilpasse den til bedriftens behov og rammer. Denne tilpasningen kalles for **Scrum-But** (Link 2). RR har tilpasset tilnærmingen ved å skille rollene ut fra det tradisjonelle prosjekt-hierarkiet og ikke ha systematiske statusmøter. Som et underordnet spørsmål til vår hovedproblemstilling ønsker vi da å vite: «Bidrar denne tilpasningen til mer effektiv prosjektstyring?»

Roller

På grunn av ressursbegrensninger har RR valgt bort Scrum-master rollen, men har heller valgt å følge den tradisjonelle rollefordelingen, ved å ha en prosjektleder. I tillegg til lederrollen har prosjektleder ved RR også rollen som utvikler da dette bidrar til effektivisering av ressursbruken. Prosjektrollefordelingen ser vi i organisasjonskartet (figur 21) som vi har utarbeidet i forbindelse med dette prosjektet. I figuren er rollene klargjort for alle som er involvert i prosjektet, inkludert prosjektlederens to roller. Under intervjuet kom det frem at prosjektlederen hadde vanskeligheter med å forholde seg til to ulike roller, da utviklerrollen gjerne hadde tidsfrister å forholde seg til og det gikk utover oppgavene han hadde i prosjektstyringen. I tradisjonell prosjektarbeid hender det at noen eller alle

teammedlemmer må pådra seg flere roller eller arbeidsoppgaver på grunn av enten kunnskap eller ressursmangel. Det hevdes å være håndterbart at å pådra seg flere roller, men rollenes ansvarsområder bør tas i vurdering. I et operativt prosjekt med tradisjonell styring kan det være mulig med flere roller for en person. Dette fordi tradisjonelle prosjekter baserer seg på en lineære strategi hvor det ikke forekommer hurtige endringer under utviklingen. I motsetning til agile prosjekter som konstant håndterer designendringer da hele utviklingsprosessen baserer seg på løpende kundetilpasning.

Usystematiske oppfølgingsmøter

I følge Scrum-metoden skal det utføres systematiske møter som Scrum- og sprint møter med hele teamet i fellesskap. Scrum-møtene skal skje daglig og har hovedhensikten med å holde den daglige fremdriften oppdatert. Sprint-møtene skal hovedsakelig brukes til å gjennomgå og planlegge neste sprint i følge Scrum-teorien. På slutten av hver sprint skal det være et sprint-review møte. I dette møtet skal den foregåtte sprinten evalueres og deretter planlegges en ny sprint. Dersom målet for en sprint var å ha klar et delprodukt skal det presenteres for produkteier, kunde og eventuelt andre av prosjektets interessenter.

I følge RRs plan skulle det holdes sprintmøte hver andre uke, hvor hovedfokuset var å planlegge neste sprint. Under intervjuet viser det seg at RR benyttet seg verken av Scrum- eller sprint møtene systematisk. RR mente at de daglige Scrum-møtene var ineffektive da møtene gjerne tok mer tid enn det som var hensiktsmessig (10-15 min) når hele teamet var samlet. Prosjektleder omformet derfor Scrum-møtene til en-til-en interaksjon med hver utvikler og det ble satt fokus på deres ansvarsoppgaver. Omformingen var for å effektivisere tidsforbruket for andre utviklere da alle ikke trengte å være på møtet i fellesskap. Sprintmøtene var så å si ikke eksisterende i RR. Møtene i Scrummetodikken ser ut til å være oppfølgingen av et prosjekt; daglige statusoppdateringer, planlegging av neste del og evaluering av foregått periode. Det er som tradisjonell prosjektgjennomføring, bare istedenfor å sette alt i en lang periode skal dette utføres flere ganger med kortere perioder. RR løste problemet med ineffektive daglige møter ved bare å ta enkelte statusoppdateringer og oppfølging med hver utvikler. Men uten å ha møte i fellesskap er det vanskelig for utviklere å holde

følge med hverandre i fremdriften. Her må prosjektleder i så fall være dyktig på å holde utviklerne oppdatert via møtene og holde lette og tilgjengelige oversikter over fremdriften. Utviklerne må også være gode på å holde hverandre oppdatert, spesielt de som jobber med oppgaver som har et avhengighetsforhold til hverandre. Det kom midlertidig frem under intervjuene at det var lite oversikter over fremdriften, og RR innrømmet at det var noe de kunne bli bedre på.

Kundeinvolvering

Scrum-metodikken tilsier at kunde og/eller produkteier bør være en aktiv deltaker i gjennomførelsen av utviklingen for at prosjektet skal ha en løpende oppdatering. I prosjektet Mobile Ticket er RR imidlertid produkteieren da de har valgt å ikke selge eierretten av det. Kunde blir da en part som kun kjøper produktet fra RR.

Kundeinvolveringen har vært valgfritt fra kundens side avhengig av hvor mye kunnskap kunden har om systemutvikling. I dette tilfellet av Flybuss Ekspress AS har det blitt satt opp en gruppe med fire kunnskapsdyktige personer innen IT fra deres side til å følge opp prosjektet med RR. Det har vært en større involvering fra kundens side på prosjektet Mobile Ticket sammenlignet med andre prosjekter. Likevel virker det ikke som kundeinvolveringen har vært optimal i følge Scrum-metodikken. Under intervjuet fremkom det at kunde ikke hadde kjennskap til agilt prosjektarbeid, og kunden opplyste at RR var den som tok seg av all planleggingen av fremdriftsplanen. Kunde var med i planleggingen i hva som leveres først, i uvisshet om at det sannsynligvis var produktbackloggen. Sett dette i sammenheng kan man ikke akkurat si at det har vært mangel på kundeinvolvering, men det spørs om det skulle vært en mer aktiv deltakelse fra kundens side sett i tråd med Scrum-metodikken. Det er med tanke på utredelsen av oppgaver til produktbackloggen og holde regelmessige sprint – planleggingsmøter og – evalueringsmøter sammen med kunde. Dette kan imidlertid forsvares med ressurser fra kundens side. Det er ikke alltid enkelt for en bedrift å sette av nok ressurser som kan følge med et sideprosjekt. Visstnok er dette en vanlig feil som prosjektteam gjør når de ikke følger Scrummetodikken fullt ut. Å ha kunde eller produkteier på den passive siden fører til at Scrum ikke fungerer. I referanseprosjektet er RR produkteier, men det er likevel kunde som skal ha siste året. Derfor burde de være mer involvert om de har ressurser å avse og gjøre det mulig i praksis.

5.2.2 Fremdriftsplan

Fremdriftsplan innebærer som nevnt integrert kostnads-, tids-, og ressursplanlegging. Dette er sett på som tre hovedstyringsfaktorer som skal sørge for at fremdriften av prosjektet skal utvikles i henhold til planen. Faktorene henger så godt sammen at det er nødvendig å ta hensyn til alle tre når man planlegger fremdriften. RR benytter seg av verktøyet Jira for kostnads-, tids- og ressursplanlegging. Referanseprosjektet Mobile Ticket var allerede blitt utviklet tidligere for AKT. Det RR skulle gjøre for Flybuss Express As var å tilpasse produktet etter kundens ønsker for krav og funksjonalitet.

Kostnadsplanlegging

Kostnadsestimeringen til RR baserer seg på internkostnaden i bedriften og fremdriftsplanen. I referanseprosjektet var de største kostnadspostene forbrukte timer på grunn av dårlig ressursplanlegging. Til oppfølging av kostnadene har RR registrerte timer i Jira, men det virker ikke som noen analyserer hva som har skjedd. I kostnadsplanleggingen savner man her en struktur i estimeringen og oppfølgingen. For det første kunne det se ut å være top-down estimering basert på at prosjekts kostnader ble estimert fra internkostnaden i bedriften og fremdriftsplanen som vil karakteriseres som et overordnet nivå. Men det reises tvil om det er den totale kostnaden for bedriften som fordeles til prosjektene. For det andre virker det ikke som estimatene baserer seg på tidligere erfaringer om direkte- og indirekte kostnader.

Tidsplanlegging

Under intervjuene ble det oppfattet slik at produktkravene ble dekomponert til funksjoner, user stories, deretter til arbeidsoppgaver, slik som beskrevet under kap.2.4.2 – «Dekomponering av produktkrav». Nye krav og funksjoner fra kunden under tilpasningen ble også inkludert.

Varigheten på oppgavene ble deretter estimert ut ifra oppgavens kompleksitetsgrad og plassert prioritert rekkefølge. Tidligere brukte RR planning poker for å estimere kompleksiteten på oppgavene i andre prosjekter, men det viste seg å ta for lang tid og RR hadde ikke kapasiteten til det. Siden utviklerne allerede hadde erfaring med

utviklingen av Mobile Ticket tidligere, tok de en skjønnsmessig vurdering av arbeidsoppgavene basert på erfaring. Selv timegraderingen ikke er beskrevet i den utarbeidede teorien om estimering av oppgaver, er det kun en variasjon av den relative graderingsteknikken under kap. 2.4.2. Sammenlignet med det er det avviker estimeringsteknikken til RR ikke fra teorien.

Opgaver som fikk tildelt kompleksitetsgraden *blocker* ble prioritert ved utviklingen av produktet. Disse oppgavene ser ut til å skape et avhengighetsforhold til de andre, og blocker kan betegnes som implikasjon; hvor en funksjon krever at en annen skal fungere. Blocker oppgavene ble sett på som startaktiviteter. Dette er en enkel og logisk måte å estimere tiden på og krever lite tid. På grunn av RRs enkle estimeringsteknikk og mye erfaring av lignende prosjekter unngikk de detaljert planlegging og hoppet tidlig ut fra planleggingsfasen til utviklingsfasen. Det er nettopp det som er meningen med styringsmodellen Scrum, at man jobber agilt ved å fokusere mer på produktutvikling enn detaljert planlegging.

Ved snakk om releaseplan hadde verken RR eller kunden hørt om det. De hadde kun en milepælplan å forholde seg til, som viste seg å være svært lik releaseplan, bare med forskjellige navn.

Selv om RR hadde utviklet produktet tidligere og hadde erfaring fra lignende oppgaver ble prosjektet likevel forsinket med to måneder. Dette skyldes blant annet dårlig planlegging og vag spesifisering. RR benyttet seg ikke av nettverksplanleggingsmetoder for prosjektene sine da dette var ukjent for dem. Graderingen av oppgavene ble kun gjort for å estimere tiden hvor det ble estimert mer tid for komplekse oppgaver enn mindre komplekse. En av kompleksitetsgradene *blocker* kan sammenlignes med avhengighetsforholdet, sekvensiell binding. De identifiserte oppgavene ved bruk av kun et bindingsforhold og i tillegg ble oppgavene lagt opp slik at utviklerne kunne jobbe parallelt med dem. Likevel hadde de ikke en god nok oversikt over disse relasjonene i styringsverktøyet sitt. De hadde heller ikke en klar oversikt over iterativ gjennomføring av aktivitetene noe som er vanlig i iterative styringsmodeller, som Scrum.

Selv om det var nok aktivitetsrelasjoner til nettverksplanlegging ble ikke dette identifisert. Alle oppgavene ble tildelt til ulike utviklere, men de hadde ikke en klar oversikt om relasjonsformen og avhengighetsforholdene til de oppgavene de jobbet med. De kunne bare forholde seg til hva de visste fra før og fortalt av prosjektleder. Her kunne det ha blitt gjort klar en tidsestimering av kravene ved hjelp av ILP (Integer Linear Programming) til releasedatoene, eller milpælene i RR sitt tilfelle. Ved å strukturere produktkravene og få et overordnet tidsestimat kunne RR eventuelt ha sett at de ikke kom til å rekke tidsfristen. Denne metoden ser ikke ut til å avvike mye fra CPM-metoden i tradisjonell prosjektstyring heller.

Mangel på nettverksplanlegging fører til at RR ikke kan effektivisere prosjektstyringen. Aktivitetsendringer underveis i programvareutviklings prosjekter, som er vanlig og skjer ofte, kan føre til at prosjektene bare blir mer forsinket uten at man kan gjøre noe tiltak for det. Det gjelder spesielt for aktivitetene som ligger i den kritiske veien. Det lønner derfor seg å identifisere flere avhengighetsforholdene og estimere oversikt over hvilke relasjoner aktivitetene har til hverandre.

Ressursplanlegging

Ressursene (utviklere) var involvert i flere andre prosjekter under utvikling av produktet Mobile Ticket for Flybuss Ekspress. Agil styringsmodell går ut på å respondere raskt på kundenes krav for å gjøre en optimal tilpasning og levere et tilfredsstillende produkt. Raske responderinger på kundenes krav kan være for eksempel være raske feilfiksinger, endringer av krav med tiden eller andre nye produktkrav som man ønsker å tilføye. Tilbakemeldinger fra flere kunder samtidig kan føre til at respondering til kundenes krav ikke skjer raskt nok. Dersom det for eksempel var en feilmelding som skulle fikses for et prosjekt, gikk det utover fremdriften av et annet prosjekt. Dette fordi utviklere må prioritere feilretting av funksjoner som for eksempel haster. Utviklere jobbet parallelt med oppgavene hvor det ble tenkt ut på forhånd om hvilke oppgaver som ville sperre for andre. Dette gikk da utover aktivitetene som egentlig var planlagt å utføre. Mangel eller dårlig på oversikt over tilgjengelig ressurs fører til at man utarbeider en fremdriftsplan ut fra feil grunnlag.

5.2.3 Oppfølging

KTR

KTR er forkortelsen av kostnad-, tid- og ressursstyring. Ut fra analysene som er blitt gjort ser vi at RR kan forbedre prosjektstyringen ut fra KTR. Det fremkommer ganske tydelig at referanseprosjektet manglet styring av disse tre styringsfaktorene.

Under intervjuet fremkom det at RR ikke fulgte oppfølgingskriteriene på KTR faktorene. Kostnadene ble fulgt opp ved å kun fokusere på de estimerte timene for oppgavene. Dette viser seg at RR mangler oppfølging av andre kostnader som utgjør en viktig del av prosjektstyringen. Under oppfølgingsteorien ser vi at det finnes et oppfølgings hjelpemiddel som baserer seg på både budsjetterte-, faktiske kostnader og inntjent verdi, S-kurven. RR benytter seg ikke av disse verdiene og har dermed ikke oversikt over kostnads-, budsjett- og fremdriftsavvik. Ved hjelp av kostnadsavviket kunne verdier blitt skapt i prosjektet og viset en oversikt over hva arbeidet har kostet (Karlsen, 2013). RR hadde da hatt et reelt tall å motivere utviklerne med.

Budsjettavviket forklarer hvordan virkelige kostnader har påløpt i forhold til budsjetterte. Et negativt tall hadde vist at prosjektet har brukt mer penger enn budsjettert hittil. Hadde det vært tilfellet for RR kunne de ha prøvd å avverge et større avvik. Fremdriftsavvik ville vist forskjellen mellom når arbeid i prosjektet er planlagt fullført og når det virkelig blir fullført. Et negativt avvik betyr at det er en forsinkelse. Med visshet i kan RR regne ut hvor mye av fremdriften som er igjen og dermed vite hvor lenge prosjektet vil bli forsinket. Eventuelt sette opp ny plan for å forhindre for lang forsinkelse.

For det andre ga RR inntrykk av at deres tidsplanlegging var for stram og ikke fungerte optimalt. Under intervjuet kom det frem at det kan hende at det bare var dagligleder eller prosjektlederen som satt alene med fremdriftsplanleggingen. Dårlige estimeringer har derfor blitt et resultat av dette da estimeringene ikke ble diskutert med andre utviklere. Det viste seg også at prosjektlederen ikke hadde all oversikt over tilgjengelige for å nå aktivitetene i milepælene i tide. Dette resulterte med at RR ikke valgte å følge prosjektet opp med det som ble budsjettert da planen var basert på feil grunnlag. RR hadde tidligere benyttet seg av hjelpemiddelet burndown chart, men den

falt bort på grunn av at det var for tidskrevende. I dette prosjektet kunne de bruke for fremdriftsavvik, men da ville de mangle IV og PV for å beregne det. Burndown chart kan dessuten sammenlignes med en S-kurve, men skiller seg ved at den viser S-kurven den andre veien, altså omvendt S-kurve.

Jira

RR ser ut til å benytte seg flittig av Jira, men de tar ikke ut det fulle potensialet til verktøyet. Jira, som er levert av Atlassian Tools, er spesialdesignet for prosjektgrupper som jobber i Scrum. Programmets funksjoner er tilpasset Scrumbrukeres behov ved å være tilgjengelig med blant annet burndown chart, velocity chart, sprint report, epic report, version report, control chart og cumulative flow diagram.

Under intervjuet fremkom det at RR benytter Jira per i dag kun for timeregistrering på oppgavene, og oppretting av sprint og dets oppgaver. RR har tidligere benyttet seg av burndown chart for å følge opp forbrukte timer i en sprint. Riktignok er metoden til burndown chart kun korrekt dersom opprinnelig anslag av sprintens timekonsum er riktig, noe som det i praksis ikke gjør i komplekse oppgaver. Men dette kunne vært et verktøy til oppfølging av hvor mange timer som er overbrukt, og deretter forbedre estimeringen for neste sprint. RR hevdet at det ville tatt for lang tid, og valgte derfor å se bort ifra denne omgangen. For å få til bedre styring bør de legge mer arbeid i oppfølgingen, og optimalisere bruken av tilgjengelige oppfølgingsmidler.

Fremgangsoversikt

I intervjuene av utviklerne fikk undertegnede inntrykk av at de savnet en overordnet oversikt over fremgangen av prosjektet. I følge Pichler (2010) kan man ha et overordnet oversikt over fremgangen ved å ha et tavle, hvor alle i utviklingsteamet kan se en overordnet oversikt over fremgangen i prosjektet.

Det fremkom at RR hadde denne oversikten i systemet i Jira. Her kunne utviklerne ved å logge seg inn, finne oversikten over prosjektet og dets fremgang. Selv om det er enkelt å gå inn i systemer for å finne informasjonen, vil noen hevde at det er enda enklere å bare kaste et blikk på en tavle. Andre vil si at det er en ”gammeldags” måte

å gjøre dette på. Men essensen i dette er å finne en enkel og motiverende måte å vise fram en overordnet oversikt på, og hvorfor da ikke gjøre det på den ”gammeldagse” måten dersom det kan fungere. Problemet her er at det kan oppstå mye arbeid når det kommer til oppdatering eller ny informasjon. Man trenger da kanskje ikke å ha en tavle, men en elektronisk skjerm som man enkelt kan sette inn data av oppdatering. Poenget er at den skal gi en enkel oversikt som alle i teamet kan følge med på.

Endringshåndtering

På grunn av vage spesifikasjoner fikk kunden sin egen tolkning av hvordan spesifikasjonene skulle være. Det viste seg at kunden forventet mer enn det som ble inngått i avtalen. Men på grunn formålet med bruk av Scrum, kundetilpasning, valgte RR å akseptere ønskene. For å leve opp til ønskene og avtalen de hadde inngått fremkom det under intervjuet at RR måtte inngå sporadiske muntlige avtaler etter krav fra kunden om fortløpende endringer underveis. I følge Scrum metodikken skulle disse endringene først kommet etter sprint evalueringsmøtene og ikke midt i utviklingsprosessen i sprintene.

Under intervjuene fremkom det at utviklerne og prosjektleder bruker en skjønnsmessig vurdering til å sette estimater for tiden de bruker for oppgavene. Det kan være forståelig da det er gjerne nye funksjoner som de ikke har jobbet med tidligere og kan ikke gjøre noe annet enn å basere seg på erfaring og ”gjetting”. Som følge av dette har det ført til feil estimater og det har vært en av årsakene til forsinkelse av prosjektene i det siste. Det gjaldt også referanse prosjektet som ble forsinket med to måneder. Utviklerne har uttrykket de sporadiske kravene som kunde har kommet med også har vært en av årsakene til forsinkelse.

En av fordelene med den agile arbeidsmetoden er å ha løpende kundetilpasning. Det er en av grunnene til at det jobbes i iterasjoner. Under intervjuene hevdet RR at det kan være uspesifiserte spesifikasjoner i avtalen som bidrar til kundens tolkninger av spesifikasjonene. Det er mulig det er en av grunnene, men poenget med løpende kundetilpasning er å kunne tilpasse kundens ønsker etter hvert som ting utvikler seg under prosessen. Kunde krav som kommer under utviklingsprosessen er en del av arbeidet. RR bør kanskje revurdere sin strategi om å ta imot tilbakemeldinger fra

kunden. Istedenfor å ta det sporadisk kan det være viktig å forberede seg for tilbakemeldinger når de sender ut demonstrasjoner, eller har møte med kunden.

En annen faktor som kom frem under intervjuene var erfaringsoverføring. Det er faktisk svært viktig med tanke på å utdanne nye roller innenfor bedriften som er så ung. Mobile Ticket var et prosjekt som hadde blitt utviklet i en annen versjon for en annen kunde tidligere. RR hadde altså erfaring med det, men det var under ledelse av en annen prosjektleder. Kvalitetssikring og erfaringsoverføring kan forbedres på dette området da prosjektleder i den nye versjonen av Mobile Ticket mente han hadde lite erfaring med prosjektstyring fra før. En idé hadde vært å avlegge god dokumentasjon av prosjektene som har vært gjennomført.

Evaluering

Under spørsmålene om evaluering i intervjuet var det noen svakheter ved styringen som gikk igjen om hva som kunne ha vært forbedret under prosjektet både hos RR og kunden. Uspesifisert spesifisering av produktkravene og dårlig prosjektrammeverk var svakhetene ved styring som alle intervjuobjektene nevnte. Et prosjektrammeverk sees på som fundamental i et prosjekt. Rammeverket skal nemlig inneholde planlagt ressursbruk og hva som er uakseptabelt tiltak og løsninger. Se utredningen i kapittel 2.1 for prosjekt mandat. Reglene gjelder da både for kunde og utviklerne, om hva som kan og ikke kan aksepteres. Det hadde muligens vært til fordel for RR som ønsker en mer strukturert løsning for kundetilpasningen. Samme gjelder for de vage spesifiseringene som gikk igjen både hos kunde og RR.

Kunden var vant til å jobbe med tunge dokumentasjoner som RR ikke hadde, og etter vår mening trenger ikke RR å ha det heller. Tunge dokumentasjoner ville ha forhindret den agile, eller smidige prosessen som de nå jobber med. Men man burde ha noe dokumentasjon. Lett dokumentasjon som går mer spesifikt i utredningen hadde vært noe å begynne med.

Ukentlige statusmøter var en forbedringsfaktor som kunde nevnte og som de nå tar i bruk i det etterfølgende prosjektet. Han synes nå at oppfølgingen er blitt mye bedre.

Punkter som bedre rollehåndtering, helhetlig oversikt av fremgangen i prosjektet, vage spesifikasjoner, muntlige avtaler med kunde og erfaringsoverføring ble også gjennomgått under evalueringen og som vi har gjort rede for tidligere i denne analysen. Bedre prosjektstyring var en faktor som ble nevnt, som egentlig oppsummer det meste av det vi har gjort rede for i denne analysen.

5.3 Red Rocks forbedringspotensial

RR er, som nevnt i innledning, et ungt selskap og har per i dag vært i drift i ca 4 år. Selskapet består hovedsakelig av ansatte med IT-utviklingsbakgrunn på både bachelor og master nivå. Selv om Scrum systemutviklingsmetoden ikke blir fulgt hundre prosent for prosjektene de har jobbet med har de likevel greid å levere et vellykket produkt til kundene sine i flere år. Det er likevel noen prosesser i denne arbeidsmetodikken som kan forbedres og bidra med å forhindre forsinkelser og øke kostnadseffektiviteten som RR har slitt med i Mobile Ticket prosjektet.

5.3.1 Forbedringsfaktorer

Ut fra intervjuene og undersøkelsene har vi kommet frem til faktorer som vi mener kan ligge til grunn for svikt i effektiv prosjektstyring i fremdriftsplanen. Disse er rangert punktvis med anbefaling av tiltak.

Designmodellen

RR jobber agilt ved å respondere raskt til kundene sine, men dette går utover utviklerens fritid da noen raske responderinger krever overtidsjobbing. Dette skyldes stor sett manglende detaljerte spesifikasjoner. Vi vil derfor anbefale RR en styringsstruktur som er litt av tradisjonelle metoder hvor man fokuserer mer på spesifikasjons- og planleggingsdelen.

RR bør optimalisere bruken av det beste av både det tradisjonelle og agile metoder. Hybrid utviklingsmetode er en blanding av tradisjonelle og agile metoder hvor man ønsker å ta i bruk det beste av begge metodene (Baird og Riggins, 2012). I stedet for å hoppe inn i sprinter velger man først å ha med seg tradisjonelle metoden i den første

sprinten som går ut på å utvikle prosjekt plan og en presentasjon som skal gis ut på slutten av sprinten. Fordelen med en slik blanding er å gi en sterk grunnlag med riktig retning for de andre sprintene i prosjektet. I tillegg vil man redusere uforutsigbarhet i prosjektet. Man kan for eksempel utarbeide kravspesifikasjon i iterasjoner, som varer over 2 måneder, men dette vil da ikke sees på som et sprint på grunn av sin lengde. Testing og feilretting av selve systemet kan for eksempel utføres i sprinter og slik kan man anvende det beste av hver metode.

Fremdriftsplan

RR benytter seg ikke av en releaseplanen per i dag. De benytter seg kun av sprint planning som utarbeides mot releasedatoen. Ved bruk av releaseplanen kan de ha mer overordnet oversikt over alle sprintene en releaseplan inneholder. Dette vil også bidra med å planlegge mer langsiktig og dermed redusere tidsrisikoen.

Kostnadsplanlegging

I kostnadsplanleggingen til RR ser det ikke ut til å være mye styring og det finnes knapt oppfølging. Vi forslår at RR burde ta for seg et generelt utviklingsprosjekt, ta for seg totale kostnader som består av indirekte og direkte kostnader. Deretter se på kostnadsnedbrytningen - Cost Break Down Structure (CBS). På den måten kan bedriften få resultatet for kostnad pr.tidsenhet og det vil gi et bedre estimat i kostnadsplanleggingen. Tiltak som anbefales i oppfølgingen her er å ta i bruk S-kurven, da den både kan brukes til kostnad – og time oppfølging. Da er det enklere for RR å se timer og kostnad forbrukt i henhold til planen og hvor langt man er i fremdriften. Det kan bli et effektivt styringsverktøy som gir en god oversikt.

Tidsplanlegging

RR bør identifisere ikke bare en, men flere avhengighetsforholdene mellom oppgavene sine for å kunne beregne seg frem til den kritiske stien. Dette er nødvendig for å fokusere mer på aktiviteter som er på den kritiske stien ved å bruke ressursene fra aktiviteter som har flyt/slakk. Dette vil bidra til å effektivisere releaseplanen. Ved hjelp av denne oversikten kan RR replanlegge releaseplanen slik at de holder

tidsfristen som er blitt lovt til kunden. Ved hjelp av nettverksplanlegging kan RR hente inn tid som har gått tapt på grunn av for eksempel feil tidsestimering.

RR bør ta i bruk mer avanserte styringsverktøy som kan hjelpe dem å fokusere på de rette aktivitetene for å nå fremdriftsmål – kritiske aktiviteter og utnytte slakk i aktivitetene til forskyvning av ressursbruk ved ressurskonflikter med andre prosjekt/oppgaver. Dette er et plan for å kunne møte kundenes krav om at leveringene skjer som avtalt. Ved hjelp av metoden kan RR hente inn tid som har gått tapt på grunn av for eksempel feil tidsestimering.

Ressursplanlegging

RR har hatt andre prosjekter som de har jobbet parallelt med prosjektet Mobile Ticket. Dette innebærer en god oversikt over hvilke ressurser som er tilgjengelig til hvilken tid. RR bør satse på mer oversiktlig overordnet plan ut i fra hva som faktisk er tilgjengelig av ressurser. For å effektivisere ressursplanlegging kreves det at man vurderer tilgjengelige ressurser før man utarbeider en fremdriftsplan slik at prosjektet får mest mulig igjen for ressursene. Både utviklere og prosjektlederen vil få nytte av å inkludere ressurser i fremdriftsplanen. Utviklerne vil få vite om når og hva de skal jobbe med (hvilket prosjekt/arbeidsoppgaver) og prosjektlederen lettere se hvor for eksempel overbelastningen av ressurser ligger. Ved overbelastning må slakk i de ulike prosjektene utnyttes, og da må planene være oppdaterte på kritisk vei.

Endringshåndtering

Faktoren med uspesifiserte spesifikasjoner kan forbedres ved å involvere flere personer. RR hevder at det kan hjelpe dersom flere utviklere er med på å spesifisere dem før de avgis til kunde. Det bør minst være to personer, gjerne utviklere som skal være med i utviklingsprosessen, eller prosjektleder og utvikler(e). Med flere personer i utarbeidelsen av spesifikasjoner er det mindre sannsynlighet for å uteglemme noe eller estimere feil. RR kan også forsøke å kalle inn til et møte med kunden og forklare

spesifikasjonene. Slik at dersom kunde har spørsmål er det enklere for dem å få svar og man kan unngå misforståelser. På denne måten om kunde kommer med endringer senere kan være enklere å håndtere det som et tillegg til hva som var avtalt på forhånd.

Regelmessige Scrummøter mellom team og kunde kan løse problemet med sporadiske tilpasninger som kunde krever. Da kan utviklere vente seg krav fra kunde etter disse møtene og ikke får seg uventede overraskelser. På den måten kan utviklingsteamet utføre løpende kundetilpasning og holde seg til planen. Dette er en forbedringsfaktor som anbefales RR for å blant annet forhindre forsinkelser. Disse møtene kan eksempelvis være daglige Scrummøter eller sprint-review møter. Dersom det skal være sprint-review møter anbefales det å ikke vente lengre enn en uke med møtene. Begrunnelsen for det er at under intervjuet kom det frem fra kunde at i IT-utviklingsprosesser bør man ikke vente for lenge med å vise sine kunder testversjoner av produktet, for da kan det blir ”for gammelt”.

Annet enn sprintbackloggen og produkt backloggen ser det ut til at RR ikke oppdaterer den skriftlige oppdateringen så ofte. All oppdatering foregår for det meste muntlig og kunde er ikke alltid klar over hvilke oppdateringer som er blitt gjort på backloggene. Kunde informerte at han fikk se fremdriftsplanen en gang og ikke vet om den ble oppdatert. Forbedringspotensialet er hyppige oppdatering og mer skriftlig dokumentasjon. For å holde arbeidet agilt anbefales det minimal med dokumentasjon, men noe mer kan være nødvendig for å lettere holde orden på hva som har blitt gjort og hva som gjenstår. Det vil også gjøre det enklere å følge med i fremgangsprosessen når alt er oppdatert til enhver tid, både når det gjelder sprint- og produktbackloggen, og fremdriftsplanen. Dermed når det oppstår endringer, vil all dokumentasjon være oppdatert for oppfølging.

For å ha prosjektstyringen under kontroll anbefales å ha integrerte kostnads-, tids- og ressursstyring med S-kurver og analyser. Dette verktøyet vil vise eventuelle avvik fra planen og gjøre det enklere å legge en ny plan dersom virkelige tall er langt unna det som er planlagt. Med S-kurver kan man følge opp tids- og kostforbruket og finne ut hvor i utviklingsprosessen prosjektet forbruker mest tid og kostnad. Analyse av dette kan brukes til å forbedre neste sprint eller senere prosjekter.

Siden RR fortsatt er en ung bedrift er det viktig at all erfaring overføres fra prosjekt til prosjekt, fra utvikler til utvikler innen samme felt, og i vårt tilfelle fra prosjektleder til prosjektleder. Her ligger det rom for forbedring. Under intervjuet av RR kom det frem at det er god kommunikasjon innad i bedriften. Det kom gjerne av at de er en liten bedrift, alle sitter veldig tett og de tar gjerne lunsj sammen og samtidig snakker om jobb. Uformelle samtaler under lunsjtid er en vanlig måte å dele erfaring på og det har visst seg å være populært innenfor organisasjonskultur. Men når det gjelder erfaring- og kvalitetsoverføring fra prosjektleder bør det formelt sett dokumenteres i tillegg til den frie og løse kommunikasjonen. I Mobile Ticket har det nå vært overføring fra en prosjektleder til en annen. Prosjektlederen for del 1 og 2 av leveringene har støtt på en god del utfordringer og har samarbeidet med kunden over en god periode. Dette vil være verdifull overføring av informasjon til nåværende prosjektleder.

Det anbefales RR å vurdere en fremdriftsplan med ressurser innlagt da det vil gi en god dokumentasjon på hva som var planlagt, og den vil vise hvor kunder kommer inn med sine endringer. Det vil gi RR utviklerne tid til å forberede seg for hvilke perioder det kan bli hektisk. Tiltak som dette anbefales sterkt for situasjoner som RR er oppi nå, hvor de har vanskelig for å holde oversikt over ressursene sine, slik at de kan forhindre forsinkelser.

Kapittel 6 - Konklusjon

Etter gjennomføring av denne oppgaven har vi kommet fram til at Scrum er en designmetodikk som ikke egner seg for total prosjektstyring av et prosjekt. Metoden tar ikke hensyn til å analysere kritiske aktiviteter og det viser seg å ha vanskeligheter med å holde kontroll på ressursene. Det er en idealisert form for designsmetodikk som i praksis fungerer dårlig for styring. Tradisjonelle styringsverktøy anbefales derfor for å gi god styring av et prosjekt i kombinasjon med Scrum-metodikken.

I analysen ser vi at det hevdes å være ulikheter mellom den tradisjonelle og agile metoden. Men etter sammenligningen ser vi at store deler den agile metoden egentlig bygger på den tradisjonelle, men har bare fått nye navn. Det kan se ut som en form av «keiserens nye klær».

Dersom man ser etter mer avansert styringsverktøy kan det anbefales Matrix – based planlegging for tid, kostnad og ressurs for identifisering av kritiske aktiviteter. RR benytter allerede milepælplan i sin gjennomføring. Det anbefales da å bruke CPM eller PERT som et av styringsverktøyene til å identifisere kritiske aktiviteter. PERT spesielt siden den beregner også usikkerhet av tidsestimatene.

I Red Rock sitt tilfelle har det vært mye slurving med å følge Scrum-metodikken helt ut. Ved analyse av dette viste det seg at de ser ut til å følge Scrum-But. Til tross for tilpasningen har det ikke fungert helt optimalt. Det anbefales derfor at Red Rock kan i det minste starte med å ha regelmessige statusmøter for å kontrollere fremgangen på prosjektet og ha oversikt over ressursene sine og deres arbeidsoppgaver. Deretter bør Red Rock ha en prosjektleder har tid til å følge opp KTR og foreta en eventuell analyse for videre prosjektplanlegging

Vi vil gjøre leseren oppmerksom på at konklusjonen som kommer av denne forskningsoppgaven og dens referanseprosjekt kan ikke generaliseres for alle som bruker Scrum-metodikken. Dette fordi datainnsamlingen er basert på kun et prosjekt og generalisering krever et større omfang.

Kapittel 7 – Fremtidig forskning

7.1 Hybrid metode

Hybrid utviklingsmetoden er en blanding av tradisjonelle og agile metoder hvor man ønsker å ta i bruk det beste av begge metodene. I stedet for å hoppe inn i sprinter velger man først å ha med seg tradisjonelle metoden i den første sprinten som går ut på å utvikle prosjekt plan og en presentasjon som skal gis ut på slutten av sprinten. Fordelen med en slik blanding er å gi en sterk grunnlag med riktig retning for de andre sprintene i prosjektet. I tillegg vil man redusere uforutsigbarhet i prosjektet. Man kan for eksempel utarbeide kravspesifikasjon i iterasjoner, som varer over 2 måneder, men dette vil da ikke sees på som en sprint på grunn av sin lengde. Testing og feilretting av selve systemet kan for eksempel utføres i sprinter og slik kan man anvende det beste av hver metode.

Hybrid metoden er en metode som kreves mer forskning på, men det er en interessant tanke som bør prøves ut da agile metoder kan trenge mer presise styringsverktøy. Mange bedrifter benytter seg av agile metoder i dag, det er da mulig at de har kombinert flere styringsmetoder for å ha bedre kontroll i bedriften. Det anbefales for fremtidig forskning for å finne suksessfaktorer blant de som blander agile og tradisjonelle metoder.

Man finner mange artikler hvor agile metoder først og fremst er rettet mot IT – bransjen. Det hadde også vært interessant å finne ut om det faktisk går an å bruke denne tilnærmingen i andre typer prosjekter, for eksempel i byggprosjekter.

Referanseliste

- Andersen, E., Grude, K., & Haug, T. (2009). *Måltrettet prosjektstyring*. Bekkestua: NKI forlaget.
- Baird, A., & Riggins, F. (2012, Fall). Planning and Sprinting: Use of a Hybrid Project Management Methodology within a CIS Capstone Course. *Journal of Information Systems Education*(23), pp. 243-257.
- Cohn, M. (2006). *Agile estimating and planning*. Massachusetts: Pearson Education, Inc.
- Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Boston: Pearson Education, Inc.
- Dalland, O. (2007). *Metode og oppgaveskriving for studenter*. Oslo: Gyldendal Norsk Forlag As.
- Elroy, J., & Ruhe, G. (2010, September). When - to - release decisions for features with time - dependent value functions. *Requirements Engineering*, pp. 337-358.
- Fernandez, D., & Fernandez, J. (2009). Agile Project Management – Agilism versus Traditional Approaches. *Journal of Computer Information Systems*, pp. 10-17.
- Gottschalk, P., & Gottschalk, A. (2012). *IT og Prosjektledelse*. Oslo: Akademika forlag.
- Gripsrud, G., Olsson, H., & Silkoset, R. (2004). *Metode og dataanalyse: med fokus på beslutninger i bedrifter*. Kristiansand: Høyskoleforlaget.
- Jacobsen, D. I., & Thorsvik, J. (2008). *Hvordan organisasjoner fungerer*. Bergen: Fagbokforlaget.
- Johannessen, A., Christoffersen, L., & Tufte, P. (2011). *Forskningsmetode for økonomiske-administrative fag*. Oslo: Abstrakt forlag.
- Karlsen, T. (2013). *Prosjektledelse-fra initiering til gevinstrealisering*. Oslo: Universitetsforlaget.
- Kolltveit, B., Lereim, J., & Reve, T. (2009). *Prosjekt - strategi, organisering, ledelse og gjennomføring*. Oslo: Universitetsforlaget.
- Larman, C. (2004). *Agile & Iterative Development*. Boston: Pearson Education, Inc.
- Layton, M. (2012). *Agile Project Management for Dummies*. New Jersey: Wilson and son Inc.
- Li, C., Akker, M., Brinkkemper, S., & Diepen, G. (2010, May). An integrated approach for requirement selection and scheduling in software release planning. *Requirements Engineering*, pp. 375-396.
- Lock, D. (2012). *Project Management*. UK: MPG Books Group.
- Mohammad, A. (2013, January). LOB and CPM Integrated Method for Scheduling Repetitive Projects. *Journal of Construction Engineering and Management*, pp. 44-50.
- PMI. (2008). *A guide to the Project Management Body of Knowledge (PMBOK Guide)*. Atlanta, USA: Project Management Institute, Inc.

- Rolstadås, A. (2011). *Praktisk prosjektstyring*. Trondheim: Tapir Akademiske Forlag.
- Szöke, Á. (2011). Conceptual scheduling model and optimized release scheduling for agile environments. *Journal of Information & Software Technology*, pp. 574-591.
- Thakurta, R. (2012). Project Executing Strategy Choice and its Impact in Software Projects under Requirements. *Journal of Management*, pp. 27-46.
- Wysocki, R. (2009). *Effective Project Management: Traditional, Adaptive, Extreme*. Indianapolis: Wiley Publications.
- Zsolt, T. (2012, December). Challenges of the project planning methods in the 21st century. *Problems of management in the 21st century*, pp. 46-60.

Internettlinker

Link 1: *Figur 10 – AON – nettverk*

Kilde: <http://hafeezrm.hubpages.com/hub/CASE-STUDY---Probability-in-PERT#slide933055>

Dato lest: 13.12.2013

Link 2: *Figur 20– Scrum Framework*

Kilde: http://www.softwaysolutions.com/blog/wp-content/uploads/2012/10/scrum_process_big3.jpg

Dato lest: 13.12.2013

Link 3: <http://www.agilealliance.org/the-alliance/what-is-agile/>

Dato lest: 13.12.2013

Link 4: <https://www.scrum.org/ScrumBut>

Dato lest: 13.12.2013

Link 5: <http://jira.codehaus.org/browse/XPR-335>

Dato lest 13.12.2013

Vedlegg

Vedlegg I

Intervjuguide – Prosjekt leder/Scrum Master

Problemstilling

«Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?»

Generelt

Hensikten med denne delen av intervjuguiden er å bli kjent med intervjuobjektet og selve prosjektet som skal analyseres med utgangspunktet i fremdriftsplanen.

Generelt om intervjuobjektet

- 1) Navn
- 2) Alder
- 3) Stilling
- 4) Utdanning
- 5) Ansettelsestid i Redrock
- 6) Tidligere arbeidserfaring

Generelt om X- prosjektet

- 1) Hva går prosjektet ut på?
- 2) Hva er den estimerte varigheten på prosjektet og på hvilket grunnlag har dere målt dette?
- 3) Hva er din rolle og arbeidsoppgaver i dette X prosjektet? Vil du definere det som Scrum master eller prosjekt leder? Og hvorfor?
- 4) Jobber du kun med dette prosjektet eller er du involvert i flere?
- 5) Har du arbeidet med liknende prosjekter tidligere?(App, webutvikling)
- 6) Består utviklingsteamet av ulike kompetanser? Hvilke?

Prosjektstyring

Hensikten med denne delen av intervjuguiden er å få en forståelse av hvilken definisjon intervjuobjektet har av prosjektstyringen, og hvordan de planlegger og oppfølger prosjektet i forhold til teorien vi har utarbeidet. Deretter skal dette analyseres opp mot teorien og finne ut hvor forbedringspotensialet ligger for effektivisering av prosjektstyringen.

- 1) Hva er din definisjon av prosjektstyring?
- 2) Hva er dine oppgaver i forhold til prosjektstyringen?

Fremdriftsplan

- 1) Hva er din definisjon av fremdriftsplan?
- 2) Hvordan er fremdriftsplanen for prosjektet bygd opp og hvem er ansvarlig for den?
- 3) Var du med i utarbeidelsen av fremdriftsplanen?

Nei

Ja

Skulle du ønske at du var det?

Hvilken rolle hadde du i det?

Ja, og hvorfor?

-

- 4) Hvordan er planen lagt opp for oppdateringer?
- 5) Foreligger det en kritisk vei i fremdriftsplanen? Hvordan er den eventuelt lagt opp?
- 6) Legger dere opp for optimalisering av aktivitetene i den kritiske veien?
- 7) Er det noe du mener som kan/kunne forbedres i fremdriftsplanen og hvorfor?

Kostnadsplanlegging

- 1) Hva innebærer kostnadsplanleggingen i Red Rock og hvem har ansvaret for den?
- 2) Hvilke kostnadsestimeringsteknikker benytter dere av? (Top-down, Bottom-up?)
- 3) Hvordan estimerer dere kostnadene for ressursene deres?
- 4) I grove trekk, hva var de største kostnadspostene dere hadde i prosjektet Mobile Ticket?
- 5) Hadde dere en "baseline" (strukturering av kostnadene) i prosjektet?
- 6) Hvordan fulgte dere opp kostnadene? Registrerte dere avvik? Brukte dere eventuelt noe verktøy? (Kostnadskontroll)

Tidsplanlegging

- 1) Hvilket program/metode(r) benytter dere av for tidsplanleggingen og hvordan fungerer den?(Burndown bar, burndown chart)
- 2) Har dere leveranse (release) plan? Eller er det noe som blir utarbeidet under sprintene? Hvorfor?
- 3) Hvordan fungerer ”Burndown” (prosjektnedbrytning) for dette prosjektet?
- 4) Hvor mange sprinter består prosjektet av og hvilken sprint er du mest aktiv i?
- 5) Hvordan fungerer den overordna tidsplanleggingen (milepælplanlegging)? Og hvor ofte oppdateres det?
- 6) Hvordan fungerer den detaljerte tidsplanleggingen (CPM/PERT)? Og hvor ofte oppdateres det?
- 7) Hva er den gjennomsnittlige tiden på hver sprint/leveranse på?
- 8) Hvilken sprint synes du er mest kritisk og/eller vanskelig og hvorfor?
- 9) Hvordan henger sprintene sammen? I forhold til den kritiske veien som finnes på tvers av sprintene?
- 10) Hvor mange oppgaver/user stories består hver sprint av?
- 11) Hvordan blir disse oppgavene/aktivitetene plassert i forhold til hverandre?
- 12) Hvordan estimerer dere tiden på hver oppgave (user story)?
- 13) Hva slags måleinstrumenter bruker dere på oppgave/ user story? (Eksempel: Story points eller planning poker)
- 14) Hvordan betegner dere om en oppgave er ferdig?(Prinsippet om ”done”)

Ressursplanlegging

- 1) Hvordan fordeles ressursene mellom ulike prosjekter?
- 2) Hvordan optimaliseres bruken av ressursene dere har tilgjengelig for dette prosjektet?
- 3) Hvordan blir ressursene tildelt hver sprint/oppgave/aktivitet?
- 4) Hvor ofte oppdateres statusen på ressursene?
- 5) Hvilke ressurser anser du som mest kritiske og hvorfor?
- 6) Har det vært forsinkelser/forhindringer i prosjektet på grunn av mangel på ressurser?

Ja

Hvordan har dere avdekket/dekket/håndtert dette?

Nei

-

Oppfølging

- 1) Hvilket program/metode(r) bruker dere for oppfølgingen av prosjektet og hvordan fungerer den?
- 2) Hvem har ansvaret for status rapportene? Og hvordan blir rapportene kommunisert med andre?
- 3) Hva og hvordan rapporterer du i dag? Til kunde? Til ledelse? Til teamet?
- 4) Hvor ofte oppdateres sprintbackloggen?
- 5) Hvor effektive er sprintmøtene og hvorfor?

- 6) Hvor effektive er de daglige Scrum møtene og hvorfor?
- 7) Forekommer det store endringer i produkt backloggen (fremdriftsplanen) etter sprint evalueringene?
- 8) Har du møtt på noen problemer hittil i prosjektet som kan ha forsinket/forhindret prosjekt arbeidet?
- 9) Hvordan håndterer/avdekker dere forsinkelser av kritiske sprinter/aktiviteter?
- 10) Hva er de typiske årsakene til forsinkelser/andre hindringer i prosjektet?
- 11) Hvilke tiltak brukes det for å håndtere hindringene? Er det slik at aktivitetene må replanlegges?
- 12) Hva er det mest brukte tiltaket for å få prosjektet på riktig kurs?

Evaluering

- 1) Hvor fornøyd er du med kommunikasjonen med utviklingsteamet og produkteieren under dette prosjektet?
- 2) Hvordan har du bidratt/kan bidra med å effektivisere prosjektstyringen?
- 3) Er det noe du hadde ønsket å ha mer/mindre av?
- 4) Eventuelt andre kommentarer du ønsker å tilføye med?

Vedlegg II

Intervjuguide – Prosjektmedarbeider/utviklingsteam

Problemstilling

«Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?»

Generelt

Hensikten med denne delen av intervjuguiden er å bli kjent med intervjuobjektet og selve prosjektet som skal analyseres med utgangspunktet i fremdriftsplanen.

Generelt om intervjuobjektet

- 1) Navn
- 2) Alder
- 3) Stilling
- 4) Utdanning
- 5) Ansettelsestid i Redrock
- 6) Tidligere arbeidserfaring

Generelt om X- prosjektet

- 1) Hva går prosjektet ut på?(mål)
- 2) Hva er den estimerte varigheten på prosjektet?
- 3) Hva er din rolle og arbeidsoppgaver i dette X prosjektet (utviklingsteam)?
- 4) Består utviklingsteamet av ulike kompetanser? Hvilke?
- 5) Jobber du kun med dette prosjektet eller er du involvert i flere?
- 6) Har du arbeidet med liknende prosjekter tidligere?

Prosjektstyring

Hensikten med denne delen av intervjuguiden er å få en forståelse av hvilken definisjon intervjuobjektet har av prosjektstyringen, og hvordan de planlegger og oppfølger prosjektet i forhold til teorien vi har utarbeidet. Deretter skal dette analyseres opp mot teorien og finne ut hvor forbedringspotensialet ligger for effektivisering av prosjektstyringen.

- 1) Hva er din definisjon av prosjektstyring?
- 2) Hva er dine oppgaver i forhold til prosjektstyringen?

Fremdriftsplan

- 1) Hva er din definisjon av fremdriftsplan?
- 2) Hvordan er fremdriftsplanen for prosjektet bygd opp og hvem er ansvarlig for den?
- 3) Var du med i utarbeidelsen av fremdriftsplanen?

Nei

Ja

Skulle du ønske at du var det?

Hva slags rolle hadde du i det?

Ja, og hvorfor?

-

- 4) Hvordan er planen lagt opp for oppdateringer?
- 5) Foreligger det en kritisk vei i fremdriftsplanen? Hvordan er den eventuelt lagt opp?
- 6) Legger dere opp for optimalisering av aktivitetene i den kritiske veien?
- 7) Er det noe du mener som kan/kunne forbedres i fremdriftsplanen og hvorfor?

Tidsplanlegging

- 1) Hvilket program/metode(r) benytter dere av for tidsplanleggingen og hvordan fungerer den?
- 2) Har dere en lanserings (release) plan?
 - Ja
Hvordan og hvor godt følges den?
 - Nei
Hvilken plan holder dere til?
- 3) Hvordan ser ”Burndown” (prosjektnedbrytning) ut for dette prosjektet?
- 4) Hvor mange sprints består prosjektet av og hvilken sprint er du mest aktiv i?

- 5) Hvordan fungerer den overordna tidsplanleggingen (milepælplanlegging)? Og hvor ofte oppdateres det?
- 1) Hvordan fungerer den detaljerte tidsplanleggingen (CPM/PERT)? Og hvor ofte oppdateres det?
- 2) Hva er den gjennomsnittlige tiden på hver sprint/leveransen på?
- 3) Hvilken sprint synes du er mest kritisk og/eller vanskelig og hvorfor?
- 4) Hvordan henger sprintene sammen? I forhold til den kritiske veien som finnes på tvers av sprintene?
- 5) Hvor mange oppgaver/user stories består hver sprint på?
- 6) Hvordan blir disse oppgavene/aktivitetene plassert i forhold til hverandre?
- 7) Hvordan estimerer dere tiden på hver oppgave (user story)?
- 8) Hva slags måleinstrumenter bruker dere på oppgave/ user story? (Eksempel: Story points eller planning poker)
- 9) Hvordan betegner dere om en oppgave er ferdig?(Prinsippet om "done")

Ressursplanlegging

- 1) Hvis ja, du jobber i flere prosjekter: Hvordan er ressursene fordelt mellom de ulike prosjektene? Ref. spm 5 – Generelt om prosjektet.
- 2) Hvordan optimaliseres bruken av ressursene dere har tilgjengelig for dette prosjektet?
- 3) Hvordan blir ressursene tildelt hver sprint/oppgave/aktivitet?
- 4) Hvor ofte oppdateres ressursene (påløpte/tilgjengelig)?
- 5) Hvilke ressurser anser du som mest kritiske og hvorfor?
- 6) Hvordan optimaliserer dere aktivitetene i den kritiske veien i forhold til ressursene?
- 7) Har det vært forsinkelser/forhindringer i prosjektet på grunn av mangel på ressurser?

Ja

Nei

Hvordan har dere avdekket/dekket/håndtert dette?

-

Oppfølging

- 1) Hvilket program/metode(r) bruker dere for oppfølgingen av fremdriften og hvordan fungerer den?
- 2) Hvem har ansvaret for status rapportene og hvordan blir rapportene kommunisert med andre?
- 3) Hva og hvordan rapporterer du i dag? Til kunde? Til ledelse? Til teamet?
- 4) Hvor ofte oppdateres aktivitetene og kritiske veien i sprintbackloggen?
- 5) Hvor effektive er sprintmøtene og hvorfor?
- 6) Hvor effektive er de daglige Scrum møtene og hvorfor?
- 7) Forekommer det store endringer i fremdriftsplanen etter sprint evalueringene som gjør at dere ligger etter planen?
 - o Ja
 - Hvilke?

- Nei
- 8) Har du møtt på noen problemer hittil i prosjektet, som ikke fremkommer i sprint/daglige møtene, som kan ha forsinket/forhindret prosjekt arbeidet?
- Ja
Hvilke? Og hva har tiltakene for dette vært? Har de fungert?
 - Nei
- 9) Hvordan håndterer/avdekker dere forsinkelser av kritiske sprinter/aktiviteter?
10) Hva er de typiske årsakene til forsinkelser/andre hindringer i prosjektet?
11) Hvilke tiltak brukes det for å håndtere hindringene? Hva mener du er mest brukt for å få prosjektet på riktig kurs?
12) Hvor nyttig synes du at kundeinvolveringen er i oppdateringer av fremdriftsplanen og hvorfor?

Evaluering

- 5) Hvor fornøyd er du med kommunikasjonen med utviklingsteamet og produkteieren under dette prosjektet? Er det noe du hadde ønsket å ha mer/mindre av? Eventuelt andre kommentarer du ønsker å tilføye med?
- 6) Hvordan har du bidratt/kan bidra med å effektivisere prosjektstyringen?

Vedlegg III

Intervjuguide – Productowner / kunderepresentant

Problemstilling

«Hva skiller moderne prosjektstyringsmodeller fra de tradisjonelle og gir de bedre styring?»

Generelt

Hensikten med denne delen av intervjuguiden er å bli kjent med intervjuobjektet og selve prosjektet som skal analyseres med utgangspunktet i fremdriftsplanen.

Generelt om intervjuobjektet

- 1) Navn:
- 2) Alder:
- 3) Stilling:
- 4) Utdanning:
- 5) Hvor godt kjenner du til Agile programvareutviklingsmetoden?

Generelt om Mobile –Ticket prosjektet

- 1) Hva gikk prosjektet ut på?
- 2) Når bestilte dere produktet og hva var ønsket leveringstid på denne? Ble den levert på ønsket tidspunkt?
- 3) Hva var din rolle og arbeidsoppgaver for dette prosjektet?
- 4) Har du vært involvert i andre prosjekter som Flybussen har outsourcet tidligere? Hva var din rolle da?

Prosjektstyring og arbeidsmetode

Hensikten med denne delen av intervjuguiden er å få en forståelse av hvilken definisjon intervjuobjektet har av prosjektstyringen og arbeidsmetoden, og hvordan de planlegger og oppfølger prosjektet i forhold til teorien vi har utarbeidet. Deretter skal dette analyseres opp mot teorien og finne ut hvor forbedringspotensialet ligger for effektivisering av prosjektstyringen og arbeidsmetoden.

- 3) Hva er din definisjon av prosjektstyring?
- 4) Var du involvert i prosjektstyringen for dette prosjektet?

Scrum

- 1) Hvor godt kjenner du til Scrum metodikken som Redrock bruker?
 - Kjenner ikke til det i hele tatt.
- 2) Hvor godt kjenner du til disse begrepene:
 - a. Produkt backlogg
 - b. Sprint
 - c. Sprint planlegging
 - d. Sprint evaluering
 - e. Scrum daglige møter
- 3) Hvor ofte hadde dere møte med Redrock?
 - a. Daglige
 - b. Ukentlige
 - c. Annenhver uke (sprintmøter?)
 - Sporadisk, etter hvert som de hadde noe levere. Nå må vi se hvordan produktet ser. De kjører nå ukentlige statuser.

Fremdriftsplan

- 8) Var du med i utarbeidelsen av fremdriftsplanen?
 - Nei
Skulle du ønske at du var det?
Ingen hadde erfaring med app utvikling
 - Ja
Hvilken rolle spilte du i denne utarbeidelsen?
- 9) Var det noe i fremdriftsplanen som kunne ha vært forbedret og hvorfor? Tidspunktene som ble lagt inn. Spesifikasjonene var spesifisert nok. Hva produktet skulle inneholde. Ønsket mer enn det som var avtalt.

10) Var du involvert i utarbeidelsen av produkt backloggen? I så fall deltok du i rangeringen av oppgavene?

Dersom du var med i utarbeidelse av fremdriftsplan/produkt backlogg:

11) Var du involvert i utarbeidelsen av prosjektnedbrytning for dette prosjektet? (Oppgave /aktivitetsnivå)

Ja

Hva synes du om prosjektnedbrytningen?

Nei

Om ikke, skulle du ønske at du var det og hvorfor?

Tidsplanlegging

10) Fikk dere en overordnet (milepæl) plan for dette prosjektet?

○ Ja

Var du med på utarbeidelsen av den?

○ Nei

Hvorfor ikke?

Hvilket plan forholdt dere til mht leveranser/demoer?

11) Hvor mange milepæler hadde dette prosjektet?

12) Hvor ofte eller hvor lang tid fikk dere leveranser på milepælene? Kunne dere komme eller kom dere med krav på dette?

Dersom du ikke kjenner til Scrum, hopp over spørsmål 4-6.

13) Ble det produkt backloggen oppdatert? Var det noen endringer som gikk igjen?

14) Ble oppdateringer av sprint backloggen tilsendt deg? I så fall hvor ofte?

15) Vet du hvor mange sprinter prosjektet bestod av? I så fall hvilken sprint syntes du var mest kritisk og hvorfor?

16) Hva slags oversikt fikk du av prosjektarbeidet?

a. Milepælnivå?

b. Aktivitetsnivå?

c. Funksjonsnivå?

d. Antall timer som ble arbeidet?

- 17) Fikk du oversikt over den tidsplanleggingen av prosjekt fremgangen og hvor ofte fikk du oppdatering på det?
- Overordnet på sprintnivå?
 - Aktivitetsnivå? (Detaljert tidsplanlegging)
- 18) Hvordan evaluerte du om en oppgave er ferdig eller ikke? Er det med hensyn til kvalitet eller funksjonalitet? (prinsippet om ”done”)

Ressursplanlegging

- 8) Har du kjennskap til fordeling av ressursene for dette prosjektet?
- Ja
Hvordan ble de fordelt? Hva synes du om denne fordelingen?
 - Nei
Skulle du ønske at du hadde det?
-
- 9) Fantes det optimalisering ved bruken av ressursene som var tilgjengelig for dette prosjektet som du kjenner til? (Eksempelvis om ressursene ble utnyttet full ut, som at alle utviklerne satt med oppgaver hele tiden)?
- 10) Var det ressurser du anså som kritiske og hvorfor? Hvilke var mest kritisk i følge din mening?
- 11) Har det vært forsinkelser/forhindringer i prosjektet på grunn av mangel på ressurser?
- Mangel på kommunikasjon.
- Ja Nei
- Hvordan ble dette avdekket/dekket/håndtert? -

Evaluering

- 7) Mener du at du har tilført prosjektet all nødvendig informasjon til utviklingsteamet for utarbeidelsen?
- 8) Hvem hadde du mest kommunikasjon med i Redrock? Hvor fornøyd er du med kommunikasjonen?
- 9) Hva synes du om prosjektstyringen til Redrock av prosjektet? Kunne du ha bidratt/kan bidra med informasjon til å effektivisere prosjektstyringen?

- 10) Er det noe du hadde ønsket å ha mer/mindre av? Eventuelt andre kommentarer du ønsker å tilføye?