Navn på forfattere:
Viktor Myrvang
Alexander E. Aasen

Tilnærminger til agile metoder for programvareutvikling i Telenor

Approaches to agile methods for software development in Telenor

MASTEROPPGAVE
Master i ledelse av teknologi
Trondheim, 23.05.2018

Veileder: Frode Heldal og Roar Stokken

Samarbeidsbedrift: Telenor Norge og Telenor Digital

Forord

Denne masteroppgaven er skrevet som siste del av det 2-årige masterstudiet Ledelse av teknologi ved Norges teknisk-naturvitenskapelige universitet (NTNU). Oppgaven ble gjennomført våren 2018 i samarbeid med casebedriftene Telenor Norge og Telenor Digital.

Vi vil benytte denne anledningen til å takke alle ansatte i Telenor Norge og Telenor Digital som har tatt seg tid til å bli intervjuet i forbindelse med denne oppgaven. På grunn av deres generøsitet har vi fått verdifull innsikt som vi vil ta med oss videre inn i arbeidslivet.

Vi vil også rette en spesiell takk til våre veiledere, Frode Heldal og Roar Stokken for deres verdifulle veiledning gjennom semesteret. Selv om tilbakemeldingene har gitt oss noen søvnløse netter, har utvilsomt resultat blitt bedre på grunn av dere. Sist men ikke minst vil vi takke Torgeir Dingsøyr, uten deg hadde vi nok fortsatt slitt med å forstå hva agile metodikker er.

Innholdet i denne oppgaven står for forfatternes regning.

God lesning!

Trondheim, mai 2018

Viktor Myrvang

Alexander E. Aasen

Sammendrag

Agile metodikker for programvareutvikling blir stadig mer utbredt i IT-bransjen. Kunder stiller større krav til oppdaterte sikre tjenester, og tradisjonelle utviklingsmetodikker gjør det utfordrende å utvikle programvare hurtig nok. Tidligere forskning beskriver godt fordelene ved å benytte og tilpasse agile metodikker, men sier lite om hvordan det skjer i praksis, hva som er driverne bak og forskjeller mellom ulike typer bedrifter. Formålet med denne studien er derfor å belyse hvordan dette skjer i praksis ved å studere en stor og en liten bedrifts tilnærming til agile metodikker.

Denne studien er en kvalitativ flercasestudie, hvor det er fokusert på to bedrifters tilnærming til bruk og tilpasning av agile metodikker for programvareutvikling, og hva som er driverne bak. Datagrunnlaget er hentet inn gjennom dybdeintervju med ledere og utviklere i de respektive bedriftene. Funnene viser at usikkerhet er den viktigste faktoren i begge bedrifters tilnærming til agile metodikker, men på ulik måte. I den store bedriften ble usikkerhet sett på noe som må minimeres for å unngå usikkerhet, mens i den lille ble usikkerhet sett på noe som kan håndteres fortløpende gjennom agile metodikker.

Det kommer også tydelig frem gjennom funnene at graden av formalitet i tilpasning av de agile metodene var forskjellig, noe som stemmer overens med tidligere forskning på området. Tilnærmingen til bruk av agile metodikker kan også sies å ha vært mer demokratisk i den mindre bedriften, noe som kan ses i sammenheng med formalitet i prosessen.

Forskningen bidrar til å gi en bedre forståelse av hvilke faktorer som er avgjørende når bedrifter skal tilnærme seg bruk og tilpasning av agile metodikker. Vi har også bidratt til å belyse hvilke likheter og forskjeller det er mellom de to casebedriftene, hvor tilnærming til usikkerhet ble funnet som den viktigste forskjellen i denne studien. Dette kan hjelpe casebedriftene til å bedre sin praksis, men kan også hjelpe lignende bedrifter i sin implementering av agile metodikker. Spesielt interessant er det hvordan den større bedriften har lagt til rette for en to-delt metodikk for å tilnærme seg usikkerhet og kontrollere risiko, en fremgangsmåte som kan være relevant for lignende bedrifter.

Abstract

Agile methods for software development is becoming increasingly more common in the IT industry. Increase in customer demands regarding secure and up to date software makes it hard for developers to deliver new software rapid enough using traditional methods. The benefits of using and tailoring agile methods is well described in previous research, but there is a lack of empirical research on real life cases and differences between types on companies. The goal of this study is therefore to shed light on the subject by studying both a large and a small companies' approach to agile methodologies.

This study is a qualitative multi-case study that focuses on two companies' approach to the use and tailoring of agile methodologies for software development, and the reasons behind said approaches. The empirical data in this study was gathered through several in-depth interviews with managers and developers in the respective companies. Our results show that uncertainty was the most important factor in both companies approach to agile methods, but in different ways. The large company focused on minimizing uncertainty to avoid risk, while the smaller company saw uncertainty as something that could be dealt with continuously using agile methods.

The degree of formality in tailoring of the agile methods was also quite different in the two companies, which fits previous research on the subject. The approach to the use of agile methods could be said to have been more "democratic" in the small company, which can be seen in correlation with the formality of the process.

The research in this study helps give a better understanding of which factors that are determining when companies are adopting agile methods. The research also contributes to illuminate the similarities and differences between the two cases, where approach to uncertainty was found to be the most important factor in this study. This could not only help the companies in question to better their practice, but also help similar companies in their implementation of agile methods. Especially interesting is it to see how the bigger company have managed to create a twofold methodology for approaching uncertainty and controlling risk, an approach which could be useful for a lot of similar companies.

Innholdsfortegnelse

1 Innledning 1

1.1 Bakgrunn 1

1.2 Kontekst 2

1.3 Metode 3

1.4 Problemstilling 3

1.5 Leserveiledning 4

2 Teori 5

2.1 Lean til Agil 5

2.2 Prosjektstruktur 5

2.2.1 Prosjekttyper 6

2.2.2 Prosjektfaser 7

2.2.3 Interessenter 8

2.2.4 Oppsummering prosjektstruktur 9

2.3 Metodikker for programvareutvikling 10

2.3.1 Tradisjonell fossefallsmetodikk 10

2.3.2 Agile metodikker 11

2.3.3 Agile praksiser 12

2.3.4 Scrum 15

2.3.5 Oppsummering metodikker for programvareutvikling 16

2.4 Usikkerhet 18

2.4.1 Oppsummering usikkerhet 21

2.5 Metodetilpasning 22
2.5.1 Tilpasning av agil metodikk 22
2.5.2 Dobbeltkretslæring 24
2.5.3 Fremgangsmåter 25
2.5.4 Oppsummering Metodetilpasning 27
2.6 Oppsummering teori 27
3 Metode 29
3.1.1 Problemstilling 29
3.2 Vitenskapelig ståsted 29
3.3 Forskningsdesign 30
3.3.1 Case 31
3.4 Datainnsamling og analyse 31
3.4.1 Intervju 31
3.4.2 Intervjuguide og gjennomføring av intervju 32
3.4.3 Utvalg av informanter 33
3.4.4 Litteratursøk 34
3.4.5 Dataanalyse 34
3.5 Forskningsoppleggets kvalitet 35
3.5.1 Relabilitet 35
3.5.2 Validitet 36
3.6 Etiske vurderinger 37
3.7 Forskningsoppleggets begrensninger 37
4 Resultat 39
4.1 Agil arbeidsmetodikk Telenor Digital 39

4.1.1 Hvordan jobber Telenor Digital? 40
4.1.2 Agile praksiser 41
4.1.3 Oppsummering 44
4.2 Agil arbeidsmetodikk Telenor Norge 45
4.2.1 Hvordan jobber Telenor Norge? 45
4.2.2 Agile praksiser 47
4.2.3 Squads 50
4.2.4 Oppsummering 51
4.3 Metodetilpasning i Telenor Digital 52
4.3.1 Arena for metodetilpasning 53
4.3.2 Squads 55
4.3.3 Oppsummering 56
4.4 Metodetilpasning Telenor Norge 57
4.4.1 Arena for metodetilpasning 57
4.4.2 Oppsummering 60
5 Diskusjon 61
5.1 Bruk av agile metodikker 61
5.1.1 Usikkerhetsstyring 61
5.1.2 Agile tilnærminger 63
5.1.3 Kontroll eller agilitet? 65
5.2 Metodetilpasning 66
5.2.1 Retrospektiv 67
5.2.2 Kontinuerlig forbedring 68
5.2.3 To-delt tilnærming 69

5.2.4 Formalitet og usikkerhet 71

6 Konklusjon 72

6.1 Implikasjoner 73

6.2 Videre forskning 73

7 Vedlegg 80

7.1 Vedlegg 1: intervjuguide 80

7.2 Vedlegg 2: Godkjent forenklet vurdering, NSD 82

Figur- og tabelliste

Figur 2-1: Prosjektveiviseren 7

Figur 2-2: Tradisjonell fossefallsmodell modifisert fra 10

Figur 2-3: Agil metodikk som en iterativ modell modifisert fra 11

Figur 2-4: Contingency factors som tilnærming til metodetilpasning 26

Tabell 2-1: Praksiser for å tilnærme seg usikkerhet 20

1 Innledning

Denne masteroppgaven har som formål å undersøke hvordan agile metodikker kan benyttes og tilpasses for å utvikle programvare i praksis, og hva som påvirker prosessen rundt. Agile metodikker legger til rette for å effektivt og fleksibelt kunne utvikle programvare, og har stort fokus på å levere høy kundeverdi (Highsmith og Cockburn, 2001). Disse metodikkene har som utgangspunkt at miljøet rundt programvareutvikling er uforutsigbart, dermed må også arbeidsmetodikken rundt utvikling legge til rette for å kunne reagere på endringer (Highsmith og Cockburn, 2001). Prinsipper som fleksibilitet, endringsvillighet og kontinuerlig forbedring står dermed sterkt i den agile filosofien (Dybå, 2009, Jørgensen, 2016, Rodríguez et al., 2012).

Fordelene ved å benytte agile metodikker for programvareutvikling er godt beskrevet i forskningslitteratur (Conboy, 2009, Campanelli og Parreiras, 2015), men forskning på prosessen rundt bruk og tilpasning av dem er i stor grad preget av teoretiske rammeverk. Disse teoretiske rammeverkene gir grunnlag for å forstå hvordan agile metodikker kan adopteres i et vakuum, men sier lite om hvordan det faktisk skjer i praksis og hva som er driverne bak. Det er flere faktorer som påvirker hvordan en bedrift velger å ta i bruk og tilpasse agile metodikker, og av den grunn viktig å belyse for å finne ut hvilken betydning de kan ha. Hensikten med denne oppgaven er derfor å bidra til en bedre forståelse om hva som påvirker hvordan bruk og tilpasning av agile metodikker skjer i praksis ved å studere to bedrifters tilnærming til agile metodikker for programvareutvikling.

1.1 Bakgrunn

Teknologiske nyskapninger og kunders raske adopsjon av digitale tjenester har siden 90-tallet resultert i en økende etterspørsel for hurtigere IT-leveranser, samtidig som bransjen har vært preget av store utfordringer knyttet opp mot økt kompleksitet, risiko, tids- og kostnadsoverskridelser (Schartum og Sørensen, 2014). Dette har utfordret de tradisjonelle måtene å drive programvareutvikling på, hvor planlegging, dokumentasjon, kontraktsforhandlinger, prosesser og verktøy har vært driverne for arbeidsmetodikken (Moe et al., 2008, Fowler og Highsmith, 2001). Dette har ført til at programvarebransjen har måtte endre arbeidsmetodikk for å kunne være konkurransedyktige. Gjennom å erkjenne at det er mennesket som må være den primære driveren for bedriftens utviklingen har agile arbeidsmetodikker i økende grad blitt adoptert for å løse utfordringene bransjen stod ovenfor (Versionone, 2017,

Rodríguez et al., 2012). Kjerneegenskapene ved agile arbeidsmetodikker blir av Conboy (2009) beskrevet på følgende måte:

> *«Den kontinuerlige beredskapen til agile metoder evner raskt å tilpasse seg endring, proaktivt eller reaktivt omfavne endring, og lære fra endring samtidig som det bidrar til opplevd kundeverdi (økonomi, kvalitet og enkelhet) gjennom sine kollektive praksiser og forhold til omgivelser»*

Agile arbeidsmetodikker har som utgangspunkt at miljøet rundt programvareutvikling er turbulent, og dermed må også arbeidsmetodikken rundt utvikling legge til rette for å kunne reagere på endring (Highsmith og Cockburn, 2001).

1.2 Kontekst

Telenor har siden opprettelsen i 1855 vært en betydningsfull leverandør av kommunikasjonstjenester for Norges befolkning, og i løpet av årene har de bygd opp tunge og sikre systemer som tjener formålet. I dag er Telenor en av verdens ledende mobilselskap med over 22 000 ansatte som produserer og leverer en rekke tjenester innen mobil, bredbånd og TV. En posisjon de har tenkt å forsterke ved å være foretrukket av kundene og levere etterspurte tjenester på en kostnadseffektiv måte (Telenor, 2018). For å kunne nå disse ambisjonene har Telenor måtte gjøre endringer i utviklingsprosessen sin, da deres tradisjonelle utviklingsmodell som kjennetegnes med en rigid prosess ikke lenger er konkurransedyktig for å kunne levere i takt med markedsbehovet.

Dette har ført til et behov for fleksible og tilpasningsdyktige arbeidsmetodikker som legger til rette for konkurransedyktig programvareutvikling og håndtering av usikkerhet. Telenor er en stor organisasjon som er delt i flere mindre selskap, og i denne studien kommer vi til å fokusere på to av dem, Telenor Norge og Telenor Digital. Telenor Norge er en stor bedrift med mange tusen ansatte som driver med alt fra drift og utbygning av mobilnettet til utvikling av programvare for det norske marked. I denne studien er det utviklingsavdelingene til Telenor Norge som står i fokus. Telenor Digital derimot er et forholdsmessig lite datterselskap som også driver med utvikling av programvare, men ikke bare for det norske marked. Hovedparten av Telenor Digitals oppgaver dreier seg om påloggingsløsningen deres, Connect ID som benyttes i Telenors plattformer verden over. Den avdelingen av Telenor Digital som er studert i denne oppgaven er på rundt 30 ansatte. Selv om Telenor Norge og Telenor Digital ligger under samme

konsern, opererer de i dag i stor grad uavhengige av hverandre. I denne studien kommer de to bedriftene til å bli sett på som to uavhengige bedrifter.

1.3 Metode

Denne forskningsstudien er en kvalitativ flercasestudie hvor data er innsamlet gjennom åtte dybdeintervju av avdelingsledere, prosjektledere og utviklere i Telenor Norge og Telenor Digital. Vi vurderte intervju for å være den mest hensiktsmessige måten å samle inn data på da studien har forsøkt å studere et fenomen vi på forhånd hadde lite kunnskap om. Intervjuene ble gjennomført på en semistrukturert måte, kodet etter hensiktsmessig kategorier og analysert opp mot relevant teori. For å finne relevant teori har vi brukt en ekstern veileder med bakgrunn i fagområdet og gjennomført litteratursøk. Denne teorien blir presentert i neste kapittel. For nærmere beskrivelse av metodiske valg henvises det til kapittel 3, «Metode».

1.4 Problemstilling

Formålet med denne studien er å belyse hvordan to bedrifter tilnærmer seg bruk og tilpasning av agile metodikker i praksis, og hvilke faktorer som har påvirket dette. Gjennom dybdeintervju med ulike ansatte i Telenor Norge og Telenor Digital har vi forsøkt å få innsikt i hvordan de har gått frem for å benytte og tilpasse agile metodikker for programvareutvikling, samt hva som har påvirket prosessen rundt. Med dette som utgangspunkt har vi valgt å definere følgende problemstilling:

> *Hva påvirker hvordan Telenor Norge og Telenor Digital tilnærmer seg agile metodikker for programvareutvikling?*

For å besvare denne problemstillingen har vi valgt å dele den opp i to forskningsspørsmål, ett basert på bruk og ett på tilpasning. Forskningsspørsmålene lyder som følger:

> FS1: *Hva påvirker hvordan Telenor Norge og Telenor Digital benytter agile metodikker for programvareutvikling?*
>
> FS2: *Hva påvirker hvordan Telenor Norge og Telenor Digital tilpasser agile metodikker for programvareutvikling?*

I tillegg til at vi vil besvare helt konkret hvordan Telenor Norge og Telenor Digital har benyttet og tilpasset agile metodikker, vil vi også se nærmere på hva som har vært førende for prosessen rundt.

1.5 Leserveiledning

Studien er naturlig bygd opp etter forskningsprosessen og delt inn i 7 hoveddeler. I dette første kapittelet har vi presentert litt om bakgrunn og kontekst for studien, samt metode og studiens problemstilling med underliggende forskningsspørsmål. I neste kapittel blir relevant teori beskrevet, med fokus på prosjektstruktur, agile metodikker for programvareutvikling og tilpasning av agile metodikker. I kapittel 3 vil studiens forskningsmetode bli redegjort for, samt at metodiske valg og studiens kvalitet og begrensninger vil bli nærmere forklart. Kapittel 4 vil presentere funnene fra datainnsamlingen, hvor disse er kategorisert og delt inn etter forskningsspørsmålenes tema. I kapittel 5 blir disse funnene analysert og diskutert opp mot det teoretiske fundamentet fra kapittel 2. Avslutningsvis følger en konklusjon i kapittel 6, samt en redegjørelse for studiens bidrag og forslag til videre forskning. Kapittel 7 består av diverse vedlegg.

2 Teori

I dette kapittelet vil det teoretiske fundamentet presenteres. Teorien som studien bygger på er delt inn i to hoveddeler: agile metodikker og metodetilpasning. For å kunne avdekke hvordan agile metodikker tas i bruk ser vi nærmere på hvordan lean har utviklet seg til agilt, prosjektstruktur i IT-prosjekt, metodikker for programvareutvikling, og hvordan usikkerhet påvirker programvareutvikling. I neste del presenteres relevant teori rundt tilpasning av agile metodikker. Her går vi inn på bakgrunn for metodetilpasninger, formalitet i metodetilpasning og ulike tilnærminger for tilpasning av agile metodikker.

2.1 Lean til Agil

Produksjonsplanlegging var en metode som Fredrik Taylor introduserte i 1911 for at ledelsen i bedrifter skulle planlegge hvordan produksjon kunne bli mer effektiv. Dette førte blant annet til høyere produktivitet gjennom bruk av for eksempel samlebånd, men metodikkene er mye kritisert for å ta lite hensyn til de ansatte (Nelson, 1980). På 1960-tallet foreslo Einar Thorsrud derfor at de ansatte skulle få større frihet til å påvirke sin egen arbeidssituasjon og ta ansvar for å styre tiden sin effektivt (Andersson, 2011). Han bidro med sitt arbeid til Lean-bevegelsen som har røtter fra produksjonssystemene til Toyota (Forbes og Ahmed, 2010). Prinsipper som «Total Quality Management», «Just-In-Time-Production» og «World Class Manufacturing» er nært beslektet med Lean som spredte seg som en slank industritrend på 90-tallet. Lean ble fort adoptert inn i IT-bransjen, og er nært beslektet med konseptet agil (Rolfsen, 2000). Agile metoder blir ofte forvekslet eller blandet med Lean, men er på flere måter forskjellig. Owen et al. (2006) presenterer ulike syn på forskjellene mellom Lean og agil hvor en av forskjellene er at de har ulike mål. Der Lean søker repeterbarhet, søker agil pålitelighet. Et annet syn er at Lean fokuserer på effektivitet og krever stabilitet, mens agile prosesser fokuserer på effektivitet og ser på de uunngåelige endringene som en mulighet til å forbedre levert verdi. «Agile project management» , eller agile prosjektledelse, er ikke synonymt Lean, til tross for noen felles røtter og like karakteristikker (Owen et al., 2006).

2.2 Prosjektstruktur

Programvareutvikling utføres i de fleste tilfeller gjennom IT-prosjekt hvor en gruppe mennesker blir satt sammen for å løse en tidsavgrenset oppgave som skal føre til et konkret mål (Aakre og Scharning, 2013). Strukturen på slike prosjekt varierer avhengig av hvilke

forutsetninger og krav som er førende, og har derfor stor betydning for hvilken arbeidsmetodikk som kan brukes. For å forstå hva som er med å påvirke hvilken arbeidsmetodikk som benyttes vil dette delkapittelet beskrive de ulike type IT-prosjekt, hvilke faser som de består av, og interessentene som er involvert.

2.2.1 Prosjekttyper

Det eksisterer mange forskjellige prosjekttyper innen IT, både når det gjelder hvor definert de er, størrelse, omfang og kompleksitet. Aakre og Scharning (2013) betegner de ulike typene som konkret, åpen og ad-hoc. Konkrete prosjekter er gjerne plandrevne, og kjennetegnes av at oppgaven er kjent, kunnskapen er stor, og det forventede resultatet tydelig. Både roller og ansvarsområder blir tydelig definert, sammen med tidsplaner og budsjetter. Kontrollbehovet er stort og det er dermed opp til menneskene å tilpasse seg rutiner og de systemene som foreligger. En motsetning til konkrete prosjekter er åpne prosjekter, som kjennetegnes av at prosessen er iterativ. At noe er iterativt betyr at det er noe som stadig gjentas. Dette er prosjekter uten definerte resultatmål og som innebærer mye prøving og feiling, med skiftende arbeidsformer og uklare roller. Her vil det motsatt av de konkrete prosjektene være rutiner og arbeidsformer som tilpasses etter behovene til de involverte. Disse prosjekttypene legger rammene for hvilken metodikk som kan brukes, hvor konkrete prosjekt krever forutsigbarhet er åpne prosjekt mer fleksibel. Som en mellomting beskriver forfatterne ad-hoc prosjekter, dette er mindre prosjekter som styres etter omstendigheter hvor prosjektet i ulike faser pendler mellom å være åpne og konkrete, og at det ofte jobbes i mindre tverrfaglige team (Aakre og Scharning, 2013).

2.2.2 Prosjektfaser

IT- prosjekter deles gjerne opp i ulike faser, og hvordan de forskjellige fasene defineres varierer mye fra bedrift til bedrift. For å kunne gi innsikt i hvordan disse fasene utføres tas det utgangspunkt i prosjektmodellen «Prosjektveiviseren», som er utarbeidet av Direktoratet for forvaltning og IKT (Difi, 2018). Modellen, som vist i figuren under, er rettet mot prosjektledere og prosjekteiere, og er utformet for alle typer prosjekter (Difi, 2018).

Figur 2-1: Prosjektveiviseren(Difi, 2018)

Første fase i Prosjektveiviseren er konseptfasen hvor det ofte tas utgangspunkt i et behov for endring gjennom intern effektivisering, forbedring av virksomhetens tjenester til samfunnet eller samfunnsbehov. Videre gjøres analyser og vurderinger av alternative løsninger, og deretter valg av den løsningen som fremstår som mest fordelaktig. Dersom løsningen har akseptabel lønnsomhet og måloppnåelse, konkluderes det med anbefalt oppstart av neste fase, planlegging. I planleggingsfasen utarbeides organiseringen, hvilke behov og funksjoner prosjektets løsning skal oppfylle, overordnet plan for prosjektets gjennomføring og en gevinstrealiseringsplan. Denne dokumentasjonen overleveres til prosjekteier for godkjennelse inn til neste fase gjennomføring. Fordi det kan være en eller flere gjennomføringsfaser i et IT-prosjekt, avhengig av usikkerhet og kompleksitet, vil type prosjekt og størrelse legge føringer for hvordan gjennomføringsfasen utføres, noe avsnitt 2.3 vil utdype nærmere. Difi forklarer det med følgende eksempel:

> *«Hvis man skal gjennomføre systemutvikling ved bruk av agil utviklingsmetode, vil dette legge føringer for roller, ansvar og gjennomføringsstrategi mv. I andre typer prosjekter kan gjennomføringsfasen(e) innebære å gjennomføre anskaffelser, inngå kontrakter og utføre detaljert analyse og design» (DiFi, 2018).*

Ettersom produktene eller leveransen utvikles i gjennomføringsfasen overføres disse til brukerne. Det kan være flere overleveringer i løpet av et IT-prosjekt, avhengig av hva prosjektet skal utvikle. Prosjektleder sikrer at prosjektstyret får tilstrekkelig med informasjon mot slutten av hver gjennomføringsfase, slik at de kan evaluere den aktuelle fasen og at leveransene knyttet til denne er ferdigstilt og godkjent. Prosjektstyret godkjenner neste faseplan, evaluerer den oppdaterte prosjektplanen og bekrefter kontinuerlig virksomhetsmessig forankring. Gjennomføringsfasen avsluttes ved at prosjekteier godkjenner at avslutningsaktivitetene kan starte. Aktivitetene i avslutningsfasen sikter mot å lære av prosjektets utfall gjennom refleksjon som kan brukes i fremtidige IT-prosjekt. Den siste fasen, realisere, innledes ved at prosjektet formelt blir avsluttet etter at produktet er levert som avtalt. Typiske aktiviteter i denne fasen er opplæring av brukerne i nye verktøy, etablering og oppfølging av arbeidsprosesser samt løpende forbedringer ettersom brukerne får erfaring.

Disse fasene er gjennomgående i de fleste IT-prosjekt men det er ulik fremgangsmåte avhengig av hvilken metodikk som brukes, noe som vil bli beskrevet i delkapittel 2.3 *metodikker for programvareutvikling*

2.2.3 Interessenter

Som nevnt tidligere er et prosjekt en arbeidsform hvor en gruppe mennesker blir satt sammen for å løse en avgrenset og gjerne krevende oppgave (Aakre og Scharning, 2013). Disse menneskene har gjerne ulike roller, og med ulike roller følger forskjellige oppgaver og ansvar. I tillegg til den interne prosjektgruppen er det også andre interessenter i prosjekt, som i ulik grad vil ha innflytelse i prosjektets prosesser og resultat.

Rose (2013) definerer interessenter på følgende måte: *«Interessenter er enkeltpersoner og grupper som er aktivt involvert i prosjektet eller som har interessenter som kan bli påvirket av prosjektet eller dets resultat, interessentene kan også omfatte de som kan påvirke prosjektets mål eller resultater».* Interessenter kan derfor regnes som personer eller grupper som er knyttet til prosjekt på ulike måter, og som i varierende grad vil ønske å påvirke et prosjekt. Disse blir ofte inndelt etter hvor stor innflytelse, krav og forventninger de har, og dermed i hvilken grad de bidrar i prosjektet (Friedman og Miles, 2006). Videre følger en nærmere beskrivelse av det de interessentene som er mest relevant for denne oppgaven

2.2.3.1 Oppdragsgiver

Oppdragsgiver er den som initierer et IT-prosjekt og i prosjektlitteraturen refereres det til oppdragsgiver som prosjekteier, prosjektstyret, klient, kunde, eller bestiller (Rolstadås, 2006). Oppdragsgivers rolle i et prosjekt er å identifisere problemet som skal løses og med det definere formålet med prosjektet. Tradisjonelt sett er det oppdragsgiver som utarbeider kravspesifikasjonen som er grunnlaget for leverandørens tilbud. Denne fremgangsmåten passer godt ved prosjekter som gjennomføres etter fossefallsmetoden, som forutsetter en uttømmende kravspesifikasjon (Schartum og Sørensen, 2014). Dermed er det oppdragsgiver som definerer de viktigste føringene og begrensningene for prosjektet, og som sørger for at prosjektet identifiseres med virksomheten.

2.2.3.2 Prosjektlederrollen

En prosjektleder har ansvaret for både resultatene, altså hva som produseres, og samhandlingen, som er hvordan det produseres (Aakre og Scharning, 2013). En av de viktigste oppgavene til en prosjektleder er å sørge for kvalitet og fremdrift i alle de sentrale prosessene som til sammen utgjør et prosjekt. I et plandrevet prosjekt er prosjektleders oppgave å:

> *«Sikre målstyring i henhold til effektmål og gevinstrealisering, til detaljert oppgave og ansvarsfordeling med tilhørende kontroll og evaluering av utførelsen. I tillegg skal prosjektlederen lage planer i henhold til tids- og kostnadsrammene og kvalitetsmål, sikre gode arbeidsforhold og støttefunksjoner for prosjektmedlemmene, samt sørge for nødvendig kommunikasjon internt og eksternt»* (Schartum og Sørensen, 2014).

2.2.4 Oppsummering prosjektstruktur

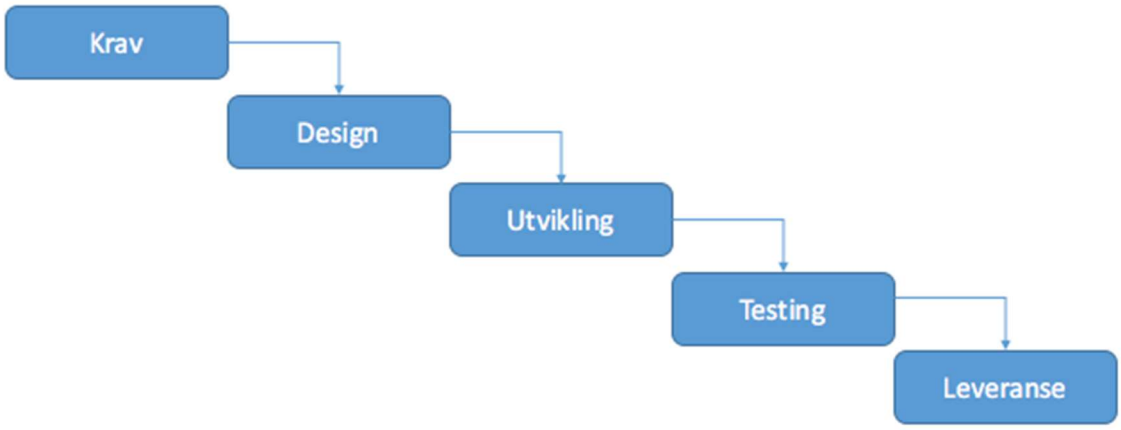
Konkrete prosjekt kjennetegnes av forutsigbarhet gjennom et stort behov for å kontrollere alle faktorene, i motsetning er åpne prosjekter uten definerte rammer hvor veien til sluttresultatet håndteres fortløpende. En mellomting mellom disse prosjekttypene er ad-hoc prosjekt som styres av omstendighetene ved å bruke mindre tverrfaglige team som veksler mellom å være konkrete og åpne. IT-prosjekt deles opp i ulike faser, men fremgangsmåten er avhengig av hvilken metodikk som benyttes og hvilke interessenter som er med på å påvirke prosjektets prosess og resultat. Interessentene i prosjektet er personer eller grupper som påvirker prosjektet, hvor oppdragsgiver er den som definerer de viktigste føringene og begrensningene for prosjektet og prosjektleder skal sørge for kvalitet og fremdrift.

2.3 Metodikker for programvareutvikling

Programvareutvikling utføres gjennom det som kan karakteriseres som to ulike paradigmer; tradisjonell metodikk og agil metodikk. Begge tilnærmingene brukes den dag i dag, men grunnlaget for agil programvareutvikling oppstod som en følge av utfordringene med tradisjonell programvareutvikling, og er derfor regnet som en forbedring på en rekke ulike måter (Fowler og Highsmith, 2001). Dette avsnittet vil derfor ta for seg tradisjonell fossefallsmetodikk, bakgrunnen og praksisene til agile metodikker, og den mest utbredte agile metoden, Scrum.

2.3.1 Tradisjonell fossefallsmetodikk

Tradisjonelt sett er kravspesifikasjonen i konkrete prosjekt formulert i detalj med kunden før utviklingen har begynt, deretter følger produktet et sett av predefinert og sekvensielle faser som ligner et fossefall som vist i figur 2-2 (Royce, 1987). Dette gjøres for å minimere endringer i prosjektet gjennom rigide forhåndsdefinerte krav som samler analyser for design og utvikling før testing av produktet utføres før leveranse. Hensikten med dette er å oppnå en høyere kvalitet ved å følge en kontrollert tidsplan (Dingsøyr et al., 2012).



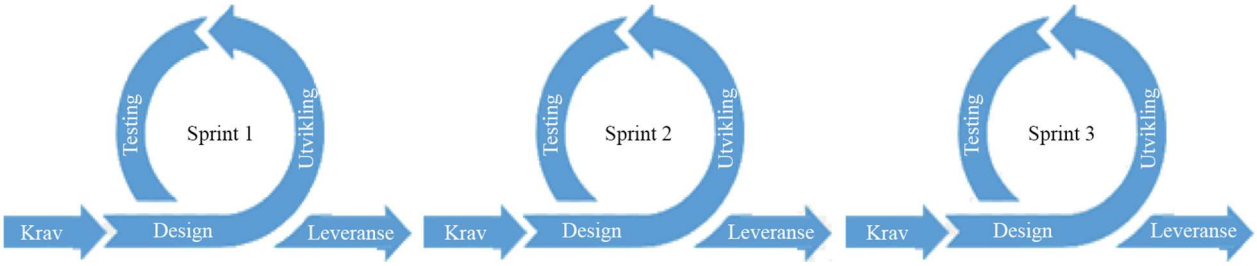
Figur 2-2: Tradisjonell fossefallsmodell modifisert fra (Royce, 1987)

Fossefallsmetodikk fokuserer mer på dokumentasjon av prosessen enn selve sluttresultatet for å sikre forutsigbarhet, men etter hvert som prosjektene blir større og mer kompleks er det vanskelig å definere alle kravene i forkant (Petersen et al., 2009). Dessuten er det avhengigheter mellom de sekvensielle fasene som gjør at metodikken ikke er særlig tilrettelagt for fleksibilitet (Moe et al., 2012). Dette fører til at mange prosjekter opplever at det er vanskelig å opprettholde tidsfristen for leveranse, tilpasse endringer som oppstår underveis og håndtere usikkerheter

(Highsmith og Cockburn, 2001). Det er derfor en stor sannsynlighet for at store forandringer må gjøres i senere faser av prosjektet, noe som kan bli kostbart og ende opp med et lite tilfredsstillende resultat (Highsmith og Cockburn, 2001).

2.3.2 Agile metodikker

I motsetning til konkrete prosjekt benytter åpne prosjekt ofte iterative agile metodikker, som vist i figur 2-3 (Ammouri, 2015). For et åpent prosjekt med agil metodikk er hver iterasjon et mini-prosjekt komponert av aktiviteter som analyse, design, programmering og testing (Larman, 2004).



Figur 2-3: Agil metodikk som en iterativ modell modifisert fra (Ammouri, 2015)

Begrepet agil metodikk ble først etablert og utviklet som et eget konsept innen programvareutvikling i 2001 da en rekke anerkjente programvareutviklere ville se om de eksisterende "lette" metodene hadde noe til felles (Cockburn, 2002). Programvareutviklerne var ifølge Cockburn (2002) uenig om det meste men klarte å enes om at det var behov for å reagere på endring, og at ordet agilt reflekterte deres intensjon. Dette resulterte i *Agile manifesto*, som i dag fungerer som et fundament for alle agile arbeidsmetodikker, hvor begrepet agil i denne sammenhengen betyr «*evnen til å skape og reagere på endring for å lykkes i et usikkert og turbulent miljø*» (Highsmith og Cockburn, 2001). Forut dette har agile metoder eksistert tidligere, men selve konseptet ble innført i nyere tid gjennom dette manifestet. Agil arbeidsmetodikk ble derfor basert på et sett verdier og prinsipper fra tidligere metoder og praksiser, hvorav agile metoder har oppstått som en følge av dette paradigmeskiftet innen programvareutvikling (Cohen, 2004). Agile metodikk er derfor basert på de fire kjerneverdiene i det agile manifest (Fowler og Highsmith, 2001), som er:

Individer og interaksjoner fremfor prosesser og verktøy

Fungerende programvare fremfor omfattende dokumentasjon

Kundesamarbeid fremfor kontraktsforhandlinger

Respondere til endring fremfor å følge en plan

Selv om de agile verdiene til venstre er høyest verdsatt, verdsettes også verdiene til høyre innenfor den agile filosofien. Agil metodikk utelater derfor ikke aktiviteter som planlegging og dokumentasjon, men holder det til et minimum. Det var ikke praksisene som var nytt med agile metoder ifølge Highsmith og Cockburn (2001), men at det er menneskene sammen med et fokus på effektivitet og manøvrerbarhet som er de primære driverne til prosjektets suksess. Agil programvareutvikling løser derfor to behov som preger dagens virksomhet og teknologi på verdensbasis: behovet for dynamiske, innovative tilnærminger og ønsket om å bygge arbeidsplasser som ikke er beskrevet i Dilbert tegneserier (Highsmith og Cockburn, 2001).

Forskningslitteratur bekrefter at agile metoder blir møtt med stor positivitet, og at brukerne trekker frem enkel adopsjon, rask utvikling, økt tilpasningsevne, læring og bedre estimering som noen av fordelene (Dybå, 2009, Jørgensen, 2016, Rodríguez et al., 2012). Det kan likevel oppstå problemer og utfordringer ved å ta i bruk agile metodikker. Tilhengere av fossefallsmetoder mener at agile metoder er kaotisk og mangler den formelle prosessorienteringen som tradisjonell tilnærming har. Agile metoder krever også styring og kontroll men i en annen form sammenlignet med de som er tilstede i tradisjonelle tilnærmelser. I stedet for kontroll- og styringsregimer, foretrekker agil metodikk lederskap og koordinering gjennom samarbeid (Nerur, 2005). Dette løser agil prosjektledelse gjennom myndige prosjektgrupper som raskt og pålitelig kan reagere på endringer og tilpasse seg etter kundens behov og miljø (Collins, 2017, Augustine, 2005).

2.3.3 Agile praksiser

Prinsippene i det agile manifest har dannet et grunnlag som sørger for at agile metoder hovedsakelig utvikles gjennom agile praksiser (Conboy, 2009). Kollektivt kan disse kombineres for å veilede programvareutvikling i et prosjekt eller en organisasjon.

I en vitenskapelig undersøkelse om bruken av Agile metoder i finsk programvareindustri, ble de følgende agile praksisene rangert som de mest brukte (Rodríguez et al., 2012):

1. Prioriterte arbeidsoppgaver
2. Iterasjoner/ sprint planlegging
3. Daglig stand-up møter
4. Enhetstester
5. Utgivelsesplanlegging

6. Aktiv kundedeltagelse
7. Selvorganiserte team (autonome team)
8. Hyppig og inkrementell leveranse av fungerende programvare
9. Automatiserte prosesser
10. Kontinuerlig integrasjon
11. Test drevet utvikling
12. Retrospektiver
13. Burn-down charts
14. Parprogrammering
15. Restrukturering
16. Kollektivt eierskap av kode

Rodríguez et al. (2012) beskriver ikke praksisene i detalj, men hevder at de kan benyttes uavhengig av hverandre. Prioriterte arbeidsoppgaver, daglig stand-ups, og retrospektiver kan for eksempel bli brukt som en del av flere metoder, men implementeres ofte forskjellig.

Disse praksisene er med på danne agile metodikker, hvor de mest utbredte metodene er Scrum og Kanban (Rodríguez et al., 2012). Kjernekonseptet med Kanban er å visualisere og kontrollere arbeidsflyten gjennom å bruke et brett bestående av kolonner og kort som representerer arbeid som skal gjøres. Hensikten er at kortene flyter uavhengig mellom kolonnene fra start frem til slutt, men det er ingen fastsatt felles standard for hvordan et Kanban-brett skal brukes i programvareutvikling (Ikonen et al., 2010). Kanban er minimalistisk i den forstand at den bare fokuserer på noen få regler for visualisering av arbeidsflyten. Den definerer ingen spesifikke roller eller arena for koordinering som gjør den enkel å implementere og adoptere (Kniberg og Skarin, 2010).

2.3.3.1 Selvorganiserte team

Agil metodikk fokuserer mindre på team bestående av spesialiserte roller og mer på selvorganiserte team som oppfordrer til rolleutveksling (Nerur, 2005). Praksisen innebærer å gi teamet autoriteten og mulighetene til å effektivt organisere og utføre deloppgaver tilpasset hver enkelt sin kompetanse og erfaring (Strand og Karlsen, 2014). Ifølge Moe et al. (2008) består fullt ut selvorganiserte team av kryssfunksjonelle generalister som kan ta på seg forskjellige arbeidsoppgaver, er allokerte og som bygger tillit og engasjement. Videre sier han at selvorganiserte team oppnår større potensielle fordeler sammenlignet med tradisjonelle team

da de bringer beslutningsmyndighet ned til operasjonelt nivå av problemer og usikkerhet, og dermed øker hastighet og presisjon til problemløsning. Denne praksisen har derfor blitt verdsatt i Scrum og det agile manifest generelt (Fowler og Highsmith, 2001). Selvorganiserte team er ikke lederløse team, eller ukontrollerte team. Lederskap i teamene er ment for å være lett og tilpasningsdyktige, ved å gi tilbakemeldinger og subtilt styre kursen for videre arbeid. Ledere av selvorganiserte team er ansvarlig for å samkjøre teamene, motivere og skaffe nødvendige ressurser, og ikke minst, vise tillit (Strand og Karlsen, 2014).

Selv om det er mange fordeler med selvorganiserte team er det en krevende arbeidsform. En studie om barrierer ved selvorganiserte team, gjort av Moe et al. (2008), viser at hovedutfordringene var å oppnå høy teameffektivitet, godt lederskap, koordinering av aktiviteter, kommunikasjon og arbeidsfordeling i forhold til de spesialiserte ferdighetene. For å unngå frustrasjon påpeker Aakre og Scharning (2013) at det er hensiktsmessig for teamet å gi hverandre innsikt i de ulike fagområdene og sørge for at alle vet hvor lang tid de ulike oppgavene tar. Å avklare de ulike avhengighetene i prosjektet, altså hvilke arbeidsoppgaver som er avhengig av hverandre, er også viktig for å unngå misforståelser og uenigheter.

Aakre og Scharning (2013) sin erfaring viser at teamdeltakere ofte er dårlige til å integrere kunnskap på tvers av fagområdene, og å ha et helhetsperspektiv gjennom prosjektet. De mener en avgjørende faktor er samarbeid, og at det er avgjørende for teamets suksess å starte den jobben i planleggingsfasen. Moe et al. (2008) påpeker videre at det er lite forskning om bruk av selvorganiserte team i agile metodikker, men at det finnes mye forskning på selvorganiserte team i seg selv. Selv om både agile metoder og Scrum baserer seg på selvorganiserende team, gis det ingen klare retningslinjer for hvordan det skal gjennomføres, da dette er tankegangen bak både agile prinsipper og Scrum.

2.3.3.2 Hyppig og inkrementell leveranse

Jørgensen (2016) fant ut at agile prosjekter som benytter inkrementelle og hyppige leveranser til kunden opplever økt sannsynlighet for at prosjektet blir vurdert som en suksess. Dette er i tråd med Koskela og Abrahamsson (2004) sine funn om at denne praksisen er nyttig for å skape fleksibilitet. Dette forutsetter at teamet indentifiserer og prioriterer funksjoner, gir tilbakemeldinger og veileder kunden gjennom hele utviklingsløpet (Vinekar, 2006). Utfordringen er at de påvirker hvordan organisasjoner må tilpasse seg for å kunne adoptere agile metoder og praksiser (Gregory et al., 2016, Javdani Gandomani et al., 2013, Nerur, 2005).

Samlet sett tyder teori på at det er en endring i konteksten hvor det er et trykk mot å levere hyppige leveranser med en kortere tid til markedet enn tidligere. Dette gjør det mulig for bedrifter å håndtere endringer raskt og holde seg konkurransedyktig.

2.3.4 Scrum

Siden opprettelsen av det agile manifest har en rekke ulike agile metoder oppstått til å veilede prosjekt på en agil måte. Ifølge den årlige undersøkelsen Versionone (2017), som tar for seg bruk av agile metodikker for programvareutvikling, brukte 70% av alle organisasjoner som deltok i undersøkelsen Scrum, med gode resultater.

Scrum er en agil metodikk og står i sterk tilknytning med verdiene til manifestet som beskrevet i avsnitt 2.3.2. Scrum benytter en iterativ, inkrementell fremgangsmåte for å optimalisere forutsigbarhet og kontrollere usikkerhet (Sutherland og Schwaber, 2017). Scrum ble, som manifestet, i utgangspunktet formalisert for utviklingsprosjekter av programvare, men det har også vist seg å fungerer godt for andre komplekse og/ eller nyskapende arbeidsomfang (Alliance, 2016). For å forklare Scrum nærmere tas det videre utgangspunkt i «The Scrum Guide», eller Scrumguiden, som er utarbeidet av grunnleggerne Sutherland og Schwaber (2017). Rammeverket har tre hovedpilarer; transparens, inspeksjon og tilpasning. Transparens betyr at de mest signifikante delene av prosessene må være synlig for de som er ansvarlig for resultatet. For å oppnå transparens må det være en felles standard for å sikre at de som trenger innsyn har sammenfallende forståelse av virkeligheten. Inspeksjon betyr at brukerne av Scrum kontinuerlig må jobbe for å oppdage avvik, og tilpasning betyr at dersom avvik registreres, må en eventuell justering skje raskt for å minimalisere videre avvik. Scrum har derfor fire ulike formelle anledninger for både inspeksjon og tilpasning: standup, sprint planleggingsmøte, sprintreview og sprint retrospektiv. Det daglige scrummøte standup er en 15 minutters samling for at utviklingsteamet kan koordinere dagens arbeid. Møtet holdes gjerne stående av den grunn at folk flest ikke liker å stå lenge, og dermed bidrar til å holde tidsrammen. Her forklarer alle medlemmene etter tur hva som er oppnådd siden forrige møte, hva som skal gjøres til neste møte, og hvilke hindringer som eventuelt har oppstått. Sprint planleggingsmøte legger planen for en sprint og utarbeides i tett samarbeid med hele scrumteamet. Teamet skal definere hva som skal leveres i løpet av sprinten, og hva som er nødvendig for å lage produktet eller inkrementet.

Sprintreview holdes på slutten av hver sprint for å gå gjennom produktet og tilpasse produktkøen. Dette møtet er uformelt, og har til hensikt å utløse tilbakemeldinger. Her snakkes det om hva som fungerte bra og mindre bra i løpet av sprinten, hvilke problemer som ble løst og hvordan. Resultatet av en Sprintreview er en revidert produktkø, eller backlog.

Sprint retrospektiv er et tilbakeblikk på den avsluttede sprinten med hovedmål om å forbedre arbeidsmetodikken før neste sprint. Her er dermed fokuset på teammedlemmene, prosesser og verktøy i stedet for produktet. Møtet identifiserer forbedringer som skal implementeres i løpet av neste sprint, og er dermed tilpasningen teamet gjør. Selv om forbedringer kan implementeres når som helst, er dette møtet en dedikert anledning for å fokusere på nettopp inspeksjon og tilpasning.

En sprint er selve «hjertet av Scrum», det er her et ferdig, brukbart og leveringsklart produkt skapes. Hver sprint er et lite prosjekt på maksimalt en måned, og består av: sprintplanleggingsmøtet, standup, selve utviklingsarbeidet, sprintreview og sprint retrospektiv. Sprinter har ofte samme varighet gjennom hele utviklingsforløpet, og en ny sprint starter umiddelbart etter at en sprint er avsluttet. Grunnen til at sprintene aldri er lengre er fordi kompleksiteten vil øke sammen med risiko. En sprint fremmer derfor forutsigbarhet, og begrenser usikkerhet. Utviklingsteamet består av fagpersoner som er nødvendig for å lage det ferdige produktet. Teamet jobber som et tverrfaglig og selvorganisert team, og det er derfor ingen som forteller utviklingsteamet hvordan de skal utvikle og ferdigstille elementene i produktkøen. Ved bruk av Scrum er dermed noen av de tradisjonelle oppgavene til prosjektlederen overført til Scrumteamet. Denne teammodellen er tatt i bruk for å optimalisere fleksibilitet, kreativitet og produktivitet. Scrum anerkjenner ingen titler i utviklingsteamet utenom utvikler, selv om teamene gjerne har spesialkompetanse eller spesielle fokusområder tilhører ansvaret teamet som helhet. Scrum betegnes som et rammeverk fordi det legges mindre vekt på spesifikke teknikker og det er mulig å implementere bare deler av den. Derfor kan Scrum ifølge (Alliance, 2016) tilpasses ulike teknikker og metoder , og samtidig fungere som er verktøy for styring og organisering. Dette presenterer Sørensen (2014) som en av årsakene til at rammeverket har fått såpass stor utbredelse.

2.3.5 Oppsummering metodikker for programvareutvikling

Programvareutvikling skjer hovedsakelig gjennom tradisjonelle og agile metodikker. Tradisjonell fossefallsmetodikk bruker et sett av predefinerte og sekvensielle faser som

gjennom analyse og dokumentasjon prøver å minimere endringer. Agil metodikk benytter en iterativ prosess som gjennom en samling av agile praksiser forsøker å skape og reagere på endring i et turbulent miljø. Prinsippene i det agile manifest har dannet grunnlaget for at agile metoder utvikles gjennom praksiser, som kombineres for å veilede programvareutviklingen i et prosjekt. Praksisen selvorganiserte team består av kryssfunksjonelle generalister som tar beslutningsmyndighet ned til et operasjonelt nivå og øker hastighet til problemløsning. Sammen med hyppig og inkrementell leveranse bidrar disse praksisene til å oppnå rask reaksjon på endring. Agile metodikker oppleves derfor som en forbedring på mange områder, men krever at organisasjonen tilpasser seg for å kunne håndtere alle hendelser i utviklingen. Den mest utbredte agile metoden er Scrum som fungerer som et rammeverk som legger til rette for de tre hovedpilarene transparens, inspeksjon og tilpasning. Dette gjøres for å optimalisere forutsigbarhet og kontrollere risiko gjennom korte sprinter bestående av en rekke teknikker og praksiser.

2.4 Usikkerhet

Programvareutviklingsbransjen vært preget av store utfordringer knyttet opp mot økt kompleksitet, risiko, tids- og kostnadsoverskridelser (Schartum og Sørensen, 2014). Disse utfordringene har fått bedrifter som bruker tradisjonelle metodikker til å innse behovet for en endring i hvordan de utvikler tjenestene om de skal være konkurransedyktig (Diallo og Patel, 2014). For å gi en innsikt i dette vil avsnittet ta for seg hvordan metodikkene tilnærmer seg usikkerhet.

Som følge av at det er vanskelig å definere alle kravene i forkant av et prosjekt, vil det være vanskelig å kunne forutse hvilke endringer som må gjøres i løpet av prosjektet. Teamets evne til å håndtere usikkerhet kan derfor være avgjørende for om et prosjekt lykkes eller ikke, og er derfor en viktig faktor for prosjektgjennomføring relatert til programvareutvikling (Gregory et al., 2015, Aakre og Scharning, 2013). Ulike tilnærminger for metodikk er viktig fordi usikkerhet er hverken iboende bra eller dårlig. Den påvirker beslutninger som kan føre til at prosjektet mislykkes, samtidig som den er et viktig aspekt for innovasjon da det ikke finnes noen økonomisk gevinst uten usikkerhet (Jalonen, 2011, Penrose, 2009).

Usikkerhet assosieres ofte med risiko, men Knight (2012) påpeker at risiko kan sannsynliggjøres, mens usikkerhet ikke kan måles. Som et resultat er det ikke mulig å tilnærme seg usikkerhet på lik måte som risiko, men prosjektledere har likevel ikke muligheten til å ignorere usikkerhet fordi det ofte fører til økte utviklingskostnader, lavere produktkvalitet, forsinket leveranse og misfornøyde kunder (Ibrahim et al., 2009). På bakgrunn av dette blir usikkerhet ofte vurdert som et negativt aspekt i litteraturen, og dens potensial for læring og innovasjon blir ofte oversett av de som opplever den (Kolltveit et al., 2004). For å unngå usikkerhet knyttet til kvalitet, tidsplan og kostnad blir grundig planlegging og kontroll over hele utviklingsprosessen anbefalt (Atkinson, 1999, Maruping et al., 2009, Fontana et al., 2015). Tradisjonell utvikling forsøker å minimere endringer gjennom prosjektet gjennom rigide forhåndsdefinerte krav som samler analyser og design. Hensikten med dette er å oppnå en høyere kvalitet ved å følge en kontrollert tidsplan (Dingsøyr et al., 2012)Agile metodikker derimot antar at usikkerhet i utviklingsprosessen ikke bare er uunngåelig, men også nødvendig og prøver å oppnå innovasjon gjennom iterasjoner som er komponert av aktiviteter som analyse, design, programmering og testing (Larman, 2004). Ved å gjøre dette brytes kompleksitet ned i mindre deler, og feil og mangler kan avdekkes tidligere i prosjektet. Når disse metodene også brukes til å la teamet respondere på uforutsette hendelser fortløpende legges det til rette for å

utvikle en programvare som til enhver tid er sammenfallende med kundes behov (Schartum og Sørensen, 2014, Highsmith og Cockburn, 2001, Highsmith, 2003, Zhiying, 2003, Moe et al., 2012). Av de grunnleggende forskjellene er prosessen i tradisjonell fossefalls metodikk basert på standard aktiviteter og målinger, i motsetning til prosessen i agile metoder som er basert på usikre aktiviteter som støtter hurtighet og kvalitet i utviklingen (Singh og Soni, 2011). Flere forskere anerkjenner derfor behovet for å håndtere usikkerhet gjennom fleksibilitet og tilpasning, i motsetning til å bare fokusere på minimering og kontroll (Moe et al., 2012, Fontana et al., 2015).

Grote (2009) presenterer to ulike tilnærminger for å takle usikkerhet; håndtering av usikkerhet og minimering av usikkerhet. Minimering fokuserer på økt kontroll og baseres på strategier som prøver å eliminere så mye usikkerhet som mulig. Denne tilnærmelsen er lite effektiv da det enten ikke er mulig eller gunstig, siden usikkerhet er uforutsigbar og læringsutbyttet er begrenset. Ved håndtering av usikkerhet forsøker man ikke å eliminere usikkerheten men tilnærme seg den fleksibelt, selv om dette krever økt innsats da den ikke følger en fastsatt prosedyre (Bannerman, 2008). Dönmez og Grote (2018) mener suksessfulle tilnærminger til usikkerhet går derfor lenger enn å bare se på konsekvensene til truslene. Det innebærer utnyttelse av fordelene til usikkerheten samtidig som man unngår eksponering av dens negative aspekter. Videre sier de at det er viktig å verifisere hvilken type usikkerhet man står ovenfor for å kunne vite hvordan man skal tilnærme seg den. De ulike type usikkerhetene er presentert i tabell 2-1 som forklarer hvilken praksis som bør brukes får å tilnærme seg ulike usikkerhetsfaktorer (Dönmez og Grote, 2018).

Tabell 2-1: Praksiser for å tilnærme seg usikkerhet (Dönmez og Grote, 2018)

Type usikkerhet	Faktor	Praksis
Ressurs	Arbeidsmengde	Oppgaver blir estimert av teamet i planlegging
	Kompetanse	Teamet har tilstrekkelig kompetanse til å utføre flere oppgaver
	Kunnskap	Sikre at prosjektinformasjon er sentralt dokumentert
	Transparens	Visualiser planlegging (tabeller, flytdiagram, kalender) basert på tilgjengelig og ikke tilgjengelig informasjon(scenariobasert)
	Matche oppgaveavhengigheter til utviklers kompetanse	Utviklere blir dynamisk tildelt oppgaver/roller for å minimere oppgaveavhengigheter (lediggang)
Krav	Manglende funksjonalitet og kvalitet	Kontinuerlig integrering av tilbakemeldinger fra kunder, ledere og andre interessenter avklarer krav og reduserer tvetydighet
	Uforventet endringer	Inkrementell planlegging og tillate endringsforespørsler på gitte tidspunkt i utviklingsprosessen, for eksempel etter en sprint
Oppgaver	Forbedre diskusjoner	Bruke kunnskapsledelse til å forbedre diskusjoner
	Uforventet oppgaver	Frigjør midlertidig medlemmer i teamet til å håndtere uforventede oppgaver
	Prosessusikkerhet	Ofte uformelle møter for å diskutere fremgang, mindre problemer, avklare oppgaver
		Refleksjoner i teamet brukes for å forbedre prosessen
	Tidsbegrensninger	Minimering av oppgavestørrelse og avhengigheter. Små oppgaver muliggjør fleksibilitet

For å kunne minimere usikkerhet hvor det er nødvendig samtidig som man håndterer den ved behov foreslår Haffke et al. (2017) Bimodal IT, en praksis som ble introdusert av Gartner (2014) hvor man benytter agil og tradisjonell fossefalls metodikk i et simultant utviklingsløp. Dette vil tilrettelegge for at fossefallsmetodikken kan fokusere på sikkerhet og stabilitet i miljø som krever forutsigbarhet, samtidig som agil metodikk kan fokuserer på fleksibilitet og hurtighet i miljø som ikke er risikoavers. Selv om det er flere definisjoner av et simultant utviklingsløp er det ingen klare beskrivelser i litteraturen som beskriver hvordan denne tilnærmingen bør skje i praksis (Horlach et al., 2016)

2.4.1 Oppsummering usikkerhet

I tradisjonell programvareutvikling brukes plandrevne modeller som fokuserer på forutsigbarhet gjennom forhåndsdefinerte krav og dokumentasjon, i motsetning bruker agil metodikk iterative modeller for å håndtere endringer i samarbeid med kunden fortløpende. Disse modellene har derfor ulik tilnærming til usikkerhet i prosjekt, hvor tradisjonell metodikk ønsker å minimere den forsøker agil metodikk å håndtere usikkerhet ved å la teamet respondere på uforutsette hendelser fortløpende. Usikkerhet har en stor påvirkning på et prosjekt, selv om de har blitt vurdert som et negativt aspekt er tilnærmelsen viktig for å kunne påvirke utfallet. Det anbefales derfor et todelt løp for å tilnærme seg usikkerhet på best måte, hvor tradisjonell fossefallsmetodikk vil minimere den i forkant og agil metodikk håndtere den fortløpende.

2.5 Metodetilpasning

Etter hvert som flere agile prosjekter mislyktes, har diskusjonen og forskning rundt tilpasning av metode for å gjøre metodikkene mer relevant for utviklingskonteksten økt . For å kunne tilnærme seg agile metodikker handler derfor en viktig del av litteraturen om å kunne tilpasse metodikkene. I dette kapittelet kommer vi først til å presentere bakgrunn for hvorfor metodetilpasning er nødvendig, for så å gå nærmere inn på de to mest brukte tilnærmingene for tilpasning av agile metodikker for programvareutvikling.

2.5.1 Tilpasning av agil metodikk

Adopsjon av nye metoder for utvikling av programvareutvikling er utstrakt (Campanelli og Parreiras, 2015, Conboy og Fitzgerald, 2010) Grunnen til dette er blant annet et ønske om å overkomme de tradisjonelle utfordringene med programvareutvikling: sene leveranser, budsjettoverskridelser og produkter som ikke leverer som lovet(Campanelli og Parreiras, 2015). Likevel argumenterer Brooks (1987) for at den konstante søken etter den «ultimate» metode kan være feilslått. All programvare er forskjellig, og ifølge Brooks (1987) vil iboende egenskaper som kompleksitet, konformitet, foranderlighet og usynlighet gjøre at programvare kan være mer komplekst enn noe annen menneskelig konstruksjon. Det argumenteres for at «there is no silver bullet»; det finnes ingen ultimat metode som passer alle mulige kontekster, men at metodene trenger høy grad av fleksibilitet for å kunne tilpasses til hvert enkelt prosjekt (Brooks, 1987). Dette støttes opp av flere (Rolland et al., 2016, Williams og Cockburn, 2003), og Conboy og Fitzgerald (2010) mener et resultat av dette er at agile metodikker sjeldent adopteres i sitt originale «tekstbokformat», men at man i praksis heller ender opp med tilpassede varianter.

Conboy og Fitzgerald (2010) påpeker flere grunner til at metodetilpasning er spesielt viktig i forbindelse med agile metodikker. På bakgrunn av det agile prinsippet "individ og interaksjon over prosess og verktøy" argumenteres det for at agile metoder må være mer fokusert på elementer som personlige karakteristikker og dynamikk enn tradisjonelle metoder må være. Videre argumenteres det for at det sterke kunde- og interessentperspektivet gjør at det ikke er nok å ta hensyn til utviklerne når man skal tilpasse metodikker, man må også legge til rette for interaksjon og samhandling utover organisasjonens grenser. Den store mengden prinsipper og verdier som er nødvendig for å «oppnå» agilitet, gjør at adopsjon og tilpasning av agile metodikker er vanskelig. Et annet argument for viktigheten av metodetilpasning av agile

metodikker, er at agilitet i sin kjerne handler om evne til å endre. Ikke bare produktet, men også måten man jobber på (Conboy og Fitzgerald, 2010). Fitzgerald et al. (2006) peker derfor på viktigheten av å skape metoder som er enkle å tilpasse ulike situasjoner, altså enkle å *endre*. Det hevdes videre at agile metodikker som er veldig deskriptive, altså består av mange faste regler og prosesser, er vanskelige å tilpasse til ulike kontekster (Turk et al., 2002, Henderson-Sellers, 2005). Kniberg og Skarin (2010) argumenterer for at hvor adaptiv en metode er i en gitt situasjon avhenger av hvor mange *regler* metoden består av. Med regler menes deler og aktiviteter metoden benytter seg av, som for eksempel artefakter, roller og aktiviteter. En stor mengde regler medfører store begrensinger, og dermed færre muligheter. Agile metodikker er i stor grad sett på som «lettvektsmetoder» som består av få regler. En av de viktigste verdiene i det agile manifest er «individ og interaksjon over prosesser og verktøy» (Fowler og Highsmith, 2001). Enkelte agile metoder er mer deskriptive enn andre, og tradisjonelle utviklingsmetoder er igjen enda mer deskriptive, noe som gjør det vanskeligere å tilpasse dem. Kanban har bare tre regler, noe som gjør at det passer godt for metodetilpasning (Kniberg og Skarin, 2010). Abrahamsson et al. (2002) argumenterer for at Scrum egner seg godt for metodetilpasning, da den består av få regler som nødvendigvis ikke må følges samt at den fungerer som et rammeverk som er tilpasningsdyktig for endringer. Dette kommer frem i litteraturen hvor forskere peker på at metoder som Scrum og Kanban har blitt brukt og tilpasset i flere utviklingsprosjekt (Fitzgerald et al., 2006, Banijamali et al., 2017, Conboy og Fitzgerald, 2010)

Pedreira et al. (2007) argumenterer for at metodetilpasning enten skjer formelt eller uformelt. I mesteparten av forskningsartiklene de tok for seg, ble det presentert en eller annen formell fremgangsmåte med regler eller retningslinjer for hvordan systematisk kunne tilpasse «off the shelf methods». Videre fant de at i praksis skjer dette som oftest på en mye mer uformell ad-hoc basis, ofte uten særlig fokus på det i det hele tatt. Generelt baserte seg man ofte på tidligere erfaringer, og tilpasset metoder på bakgrunn av det. Større organisasjoner hadde en viss tendens til å benytte seg av formaliserte prosesser, men i mindre organisasjoner skjedde dette sjeldent. Dette er de samme funnene som Conboy og Fitzgerald (2010) gjorde i sin forskning.

I forskningslitteraturen brukes mange forskjellige begreper for å definere prosessen bak tilpasning av agile metodikker, hvor «Method tailoring», «Method assembly», «Method adoption» og «Software process tailoring» er blant de mest brukte. Som tidligere nevnt er definisjonen på disse i stor grad sammenfallende, og denne rapporten kommer til å benytte «metodetilpasning» som navn på prosessen. Definisjonen på metodetilpasning er følgende *«en*

prosess eller evne der hvor menneskelige agenter gjennom responsive endringer i, og dynamisk samspill mellom, kontekst, intensjon og metodefragmenter bestemmer en programvareutviklingsmetodologi for en spesifikk prosjektsituasjon» (Seyam, 2010).

Litteraturen fokuserer i stor grad på å finne ulike fremgangsmåter for metodetilpasning. Disse har igjen mange forskjellige navn i litteraturen, men Conboy og Fitzgerald (2010) og Campanelli og Parreiras (2015) argumenterer for at disse kan kategoriseres innenfor en av to overordnende tilnærminger, «method engineering» og «contingency factors». Disse kommer til å bli videre presentert i kapittel 2.6.3.

2.5.2 Dobbeltkretslæring

Lyytinen og Robey (1999) sier at en nøkkelfaktor til vellykkede IT-prosjekter ligger i utviklingsteamets evne til å reflektere fra, og dermed lære av, tidligere prosjekt. Gjennom et stort fokus på læring og forbedring, har dobbeltkretslæring blitt mer og mer viktig i agile metodikker og metodetilpasning. Morgan (2004) sier at i dobbeltkretslæring vil en først oppfatte og registrere omgivelsene, før informasjonen som er mottatt sammenliknes med eksisterende normer. Deretter settes det spørsmålstegn ved hvorvidt handlingsnormene egner seg og så iverksettes den adferden som er mest egnet. Enkeltkretslæring vil ikke sette dette spørsmåltegnet, men gå rett på å iverksette tiltak. Her mistes derfor muligheten til å lære. Morgan (2004) hevder at mange organisasjoner er flinke på enkeltkretslæring ved at de utvikler en evne til å registrere miljøet, de setter mål og retter sine prestasjoner inn mot disse målene. Derimot er dobbelkretslæring vanskeligere å oppnå da det krever store ressurser som organisasjonene sjeldent prioriterer.

En tilnærming for å oppnå dobbeltkretslæring har vært å fokusere på retrospektiver. Nelson (2010) beskriver retrospektiv som en forhåndsdefinert metode for å evaluere prosjektets prestasjon, utvinne erfaringer og videre anbefalinger. Dette gjøres gjennom en systematisert, formalisert gjennomgang av hvilke erfaringer som kan læres fra produkter, prosesser og ressurser for å sikre fordeler i pågående og fremtidige prosjekt (Myllyaho et al., 2004). For å vurdere hvorvidt en retrospektive kan oppnå sitt mål er det viktig å vurdere hvilke faktorer som kan er hensiktsmessig å fokusere på, da feil fokus enkelt kan føre til at man bare oppnår enkeltkretslæring i retrospektivene. Myllyaho et al. (2004) har identifisert seks fordeler som oppstår ved å reflektere over aktiviteter i et prosjekt:

1. Å erverve en bedre forståelse av andre team medlemmers perspektiv

2. Å integrere individ/team læring
3. Å identifisere skjulte problemer
4. Å legge til rette for dokumentasjon av god praksis som skal repliseres og dårlig praksis som skal utelates
5. Å identifisere individuell arbeidstilfredshet som et resultat av tilbakemeldinger
6. Å forbedre estimeringer

Collier et al. (1996) foreslår ytterligere at teamets refleksjoner legger til rette for at individene får identifisert og diskutert prosjektkonflikter, og vurdert om teamsamarbeid er nødvendig. Ved å bruke disse iterative metodene kan man klare å oppnå dobbelkretslæring gjennom hele prosjektet.

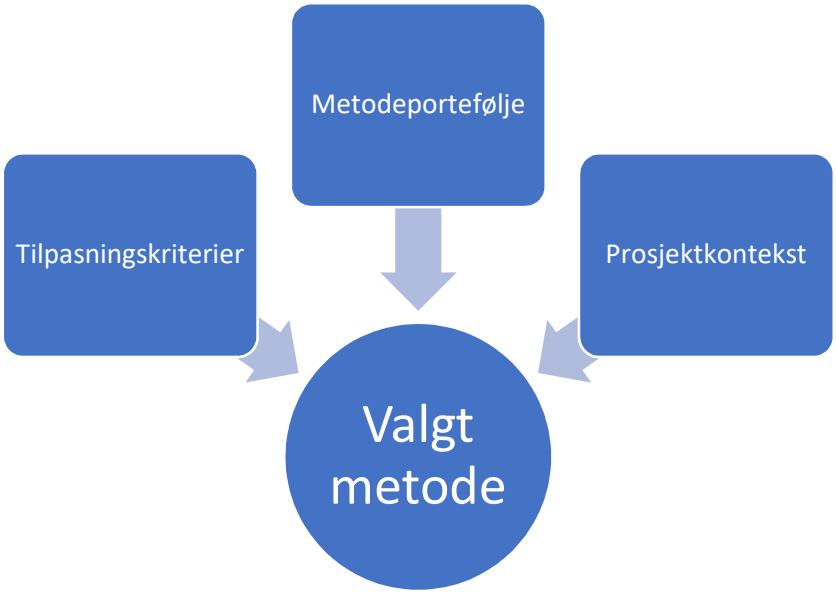
2.5.3 Fremgangsmåter

Som tidligere nevnt, fokuserer mye av forskningslitteraturen om metodetilpasning på å finne fremgangsmåter for å tilpasse agile metodikker. Dette var noe man forsket på også før de agile metodikkene vokste frem, men mengden forsknings på område har hatt en stødig vekst etter hvert som adopsjonen av agile metodikker har økt (Campanelli og Parreiras, 2015). Forskere har identifisert fremgangsmåter for metodetilpasning basert på nye empiriske studier, tidligere forskning på andre fagområder samt introdusert egne mulige varianter. Fremgangsmåtene er i stor grad sammenfallende, og som tidligere nevnt kan de ifølge Campanelli og Parreiras (2015) kategoriseres innenfor en av to overordnede fremgangsmåter: contingency factors og method engineering.

2.5.3.1 Contingency factors

Contingency factors er en fremgangsmåte for metodetilpasning som handler om å velge ut en spesifikk eksisterende metode som passer til en gitt kontekst (Iivari, 1989). Fremgangsmåten er basert på premisset at ingen enkelt metode kan passe til alle mulige kontekster, men at hvis man har mange nok metoder å velge mellom, vil man alltid finne minst én som fungerer. Sentralt for denne fremgangsmåten blir dermed at de som ønsker å bruke den må ha erfaring og kunnskap om mange ulike metoder, og evne til å se hvilken metode som passer til hvilken kontekst. Contingency factors som fremgangsmåte setter store krav til teamet, eller organisasjonen, og at man må ha høy toleranse for endringer. Hvis man har veldig ulike prosjekter kan det være krevende å måtte endre arbeidsmetode ofte. Eksempler på faktorer som kan være med på å bestemme hvilke metoder man skal velge kan være interne faktorer som

teamstørrelse eller teamdistribusjon, eller eksterne faktorer som kundeinvolvering og kontraktstype (Campanelli og Parreiras, 2015).



Figur 2-4: Contingency factors som tilnærming til metodetilpasning (Campanelli og Parreiras, 2015)

Contingency factors baserer seg som nevnt på at de som benytter det har god kunnskap om mange forskjellige metoder, samt ha evnen til å vurdere hvilken som passer i gitte kontekster. Dette blir av flere sett på som en svakhet, da dette viser seg å være vanskelig i praksis. Det blir også sett på som en svakhet at contingency factors har som utgangspunkt at det finnes en spesifikk metode til hvert mulig prosjekt (Conboy og Fitzgerald, 2010) . Dette er til en viss grad motstridende for kjerneprinsippet til metodetilpasning, som er at det ikke finne noen predefinert metode som kan dekke alle behov til et hvilket som helst prosjekt.

2.5.3.2 Method Engineering

Method engineering er definert som ingeniørdisiplinen som omhandler å designe, konstruere og adoptere metoder, teknikker og verktøy for bruk i programvareutvikling (Henderson-Sellers et al., 2014). I praksis handler method engineering om å ta i bruk metodefragment fra forskjellige metoder for så å sette dem sammen til en ny metode til en spesifikk situasjon (Rolland og Prakash, 1996). Siden dette gjøres in-house, gjør det at teamet og organisasjonen har stor fleksibilitet knyttet til å tilpasse metoden til spesifikke prosjekt (Campanelli og Parreiras, 2015). En metode blir sett på som en samling av metodefragment, og et metodefragment kan være en beskrivelse av en utviklingsmetode eller en hvilken som helst tilknyttet del. Det kan være prinsipp, fundamentale konsept, produktleveranser, aktiviteter som

må gjøres, arbeidsverktøy, teknikker, tips osv. (Aydin et al., 2004). Ifølge Campanelli og Parreiras (2015) er method engineering den mest brukte fremgangsmåten for metodetilpasning.

Fitzgerald et al. (2000) argumenterer for at hvis fremgangsmåten skal fungere, er det nødvendig med en «metodebase» og en dedikert «metodeingeniør» for å være suksessfull. En metodebase kan sees på som et lager av metodefragment, og en metodeingeniør er en person som velger de ulike metodefragmentene som skal brukes i hvert enkelt prosjekt. Fitzgerald et al. (2006) argumenterer imidlertid for at fremgangsmåten kan være suksessfull uten den dedikerte metodeingeniører eller metodebaser. I deres case ble deler av Scrum og XP kombinert, og resultatet var relativt vellykket. Mye av kritikken mot method engineering går på at det er en kompleks og omfattende prosess, og dermed kan være vanskelig å benytte i praksis for mange organisasjoner (Fitzgerald et al., 2000).

2.5.4 Oppsummering Metodetilpasning

Metodetilpasning er en fremgangsmåte for å kunne gjøre de agile metodikkene mer tilpasset for utviklingskonteksten. Dette gjøres da metodikkene i teorien ikke er tilpasset organisasjonens behov og trenger justeringer for å fungere i praksis. I større organisasjoner skjer dette ofte gjennom formelle prosesser som tar utgangspunkt i teori, mens i mindre organisasjoner benyttes gjerne en mer uformell tilnærming ved å ta utgangspunkt i erfaring. Dette kan gjøres gjennom dobbeltkretslæring hvor retrospektiver blir pekt på som en arena for å oppnå dette. Utfordringen er at agile metodikker består av en mengde prinsipper og verdier som gjør at tilpasning av metodikken kan være utfordrende. For å løse disse utfordringene har method engineering og contigency factors oppstått som to overordnede tilnærminger. Contingency factors er en fremgangsmåte for å velge ut spesifikke metode som passer til gitt kontekst og setter dermed store krav til kunnskap til å kunne vurdere hvilken metode som skal benyttes til enhver tid. Method engineering er en disiplin som designer, konstruerer, og adopterer metoder, teknikker og verktøy for bruk i programvareutviklingen.

2.6 Oppsummering teori

Programvareutvikling skjer hovedsakelig gjennom tradisjonell og agil metodikk. Tradisjonell fossefallsmetodikk bruker et sett av predefinerte og sekvensielle faser som gjennom analyse og dokumentasjon prøver å minimere endringer. Agil metodikk benytter en iterativ prosess som gjennom en samling av agile praksiser forsøker å skape og reagere på endring i et turbulent miljø. Prinsippene i det agile manifest har dannet grunnlaget for at agile metoder utvikles

gjennom praksiser, som kombineres for å veilede programvareutviklingen i et prosjekt. Praksisene selvorganiserte team og hyppig inkrementell leveranse bidrar til at programvareutviklingen blir vurdert som en suksess. Agil metodikk oppleves derfor som en forbedring på mange områder men krever at organisasjonen tilpasser seg for å kunne håndtere alle hendelser i utviklingen. Den mest utbredte agile metoden er Scrum som fungerer som et rammeverk som legger til rette for å kunne benytte en rekke praksiser og verktøy for å oppnå agilitet. Dette gjøres for å optimalisere forutsigbarhet og kontrollere risiko gjennom korte sprinter bestående av en rekke teknikker og praksiser. De forskjellige metodikkene har ulik tilnærming til usikkerhet i prosjekt, hvor tradisjonell metodikk ønsker å minimere den forsøker agile metodikker å håndtere usikkerhet ved å la teamet respondere på uforutsette hendelser fortløpende. Usikkerhet har en stor påvirkning på et prosjekt, og selv om det har blitt vurdert som et negativt aspekt er tilnærmingen til dem viktig for å kunne påvirke utfallet. En måte å tilnærme seg usikkerhet og risiko på kan være et todelt løp, hvor man benytter tradisjonell fossefallsmetodikk for å minimere usikkerhet og risiko hvor dette ikke kan tolereres, og agile metodikker for å håndtere usikkerhet hvor dette ses på som optimalt.

Agile metodikker trenger ofte tilpasning for å fungere optimalt i praksis, og metodetilpasning er derfor blitt en vanlig del av tilnærming til agile metodikker for for å kunne gjøre de agile metodikkene mer tilpasset den spesifikke utviklingskonteksten. Gjennom uformell og formell tilnærming tar organisasjoner utgangspunkt i erfaring hvor dobbeltkretslæring blir vurdert som en god prosess for å oppnå verdifull læring. Dette krever fokus og ressurser, hvor retrospektiv er en god arena for å kunne gjennomføre dette. Utfordringen er at agile metodikker består av en mengde prinsipper og verdier som gjør at tilpasning av metodikken kan være utfordrende. For å løse disse utfordringene har method engineering og contigency factors oppstått som to overordnede tilnærminger. Contingency factors er en fremgangsmåte for å velge ut spesifikke metode som passer til gitt kontekst og setter dermed store krav til kunnskap til å kunne vurdere hvilken metode som skal benyttes til enhver tid. Method engineering er en disiplin som designer, konstruerer, og adopterer metoder, teknikker og verktøy for bruk i programvareutviklingen

3 Metode

I dette kapittelet kommer studiens forskningsmetode til å bli redegjort for. Vi vil i kapittelet gå nærmere inn på vitenskapelig ståsted, forskningsdesign, datainnsamling, forskningsopplegget kvalitet, etikk og forskningsoppleggets begrensninger. Målet med kapittelet er å redegjøre for valg som er gjort, samt å vurdere kvaliteten og begrensningene til forskningsopplegget

3.1.1 Problemstilling

For å komme frem til problemstillingen, ble det gjennomført to innledende ustrukturerte intervju med to ledere i Telenor Norge og Telenor Digital. Tema i disse intervjuene var arbeidsmetodikk, og på bakgrunn av disse intervjuene ble en arbeidsproblemstilling laget med bakgrunn i temaet *agil* arbeidsmetodikk. Etter dette ble det gjennomført et innledende litteratursøk for å sette seg mer inn i fagområdet samt å finne eventuelle interessante problemstillinger. Tilpasning av agile metodikker, og forskjeller i bruk av agile metodikker i små og store bedrifter var to av temaene som ble ansett som interessant. Dette litteratursøket la bakgrunn for datainnsamlingen, som er nærmere beskrevet i kapittel 3.4.

Problemstillingen i denne studien er «*Hva påvirker hvordan Telenor Norge og Telenor Digital tilnærmer seg agile metodikker for programvareutvikling?».* Problemstillingen er åpen og eksplorerende, men avgrenser samtidig flere faktorer. Problemstillingen er åpen i den forstand at hva som menes med tilnærming er åpent for diskusjon, mens den er avgrensende i den forstand at det er «agile metodikker for programvareutvikling» i «Telenor Digital og Telenor Norge» som skal undersøkes.

3.2 Vitenskapelig ståsted

I vitenskapsteori skiller man mellom ontologisk og epistemologisk teorier. «Ontologi dreier seg om grunnleggende antakelser om hvordan den sosiale verden ser ut, mens epistemologiske teorier er ulike oppfatninger om hvordan man skal skaffe seg kunnskaper om denne verdenen» (Johannessen et al., 2011). Innenfor ontologiske teorier skiller man mellom individualisme og kollektivisme, der individualister mener at samfunnet ene og alene består av individer og deres handlinger, mens kollektivister anser samfunnet som en egen realitet som ikke kan reduseres til enkelt individers handlinger (Johannessen et al., 2011). Epistemologi omhandler i større grad hva som er «sann» kunnskap, og hvordan man tilegner seg denne. Vi anser det derfor som viktig å anerkjenne at vår virkelighetsoppfatning, vårt ontologiske standpunkt, påvirker hvordan vi

tilegner oss og vurderer kunnskap. Ifølge Johannessen et al. (2011) er dette særdeles viktig i kvalitativ forskning, der forskeren selv står for innhenting, analyse og fortolkningen av resultatene. Denne studien er derfor verken en rendyrket individualistisk eller kollektivistisk, men er heller preget av det sosialkonstruktivistisk verdenssynet, der man tror at mennesker ønsker å forstå verden rundt seg, og at deres subjektive meninger er det som ligger til grunn for deres tolkning av verden (Creswell, 2014).

I vitenskapsteori trekker man også et skille mellom hermeneutisk og positivistisk tilnærming til forskning (Johannessen et al., 2011). Positivister mener at alle fenomen burde undersøkes med samme naturvitenskapelige metode, og at sosiale fenomen kun burde studeres *utenfra*. Hermeneutikere mener på sin side at det ikke finnes lovmessigheter innen det samfunnsmessige område, og sosiale fenomen burde dermed studeres på en mer fortolkende måte (Johannessen et al., 2011). Denne studien har hatt en hermeneutisk tilnærming til forskning da vi har hatt som formål å fortolke hvordan to bedrifter har tilnærmet seg agile metodikker i praksis.

3.3 Forskningsdesign

I dette delkapittelet kommer valgene som har blitt gjort med tanke på forskningsdesign til å bli presentert.

Siden dette er en studie med hensikt om å opparbeide en dypere forståelser av et eller flere fenomen, ble en kvalitativ tilnærming vurdert som mest hensiktsmessig. Det som skiller kvalitativ metode fra kvantitativ, er at empirien baserer seg på ord i stedet for tall. Dette er ifølge Johannessen et al. (2011) hensiktsmessig hvis man skal undersøke et fenomen man ikke kjenner spesielt godt, eller oppnå en dypere forståelse av et fenomen.

I forskningslitteratur skilles det også mellom deduktiv og induktiv tilnærming til forskning, der deduktiv handler om å gå fra «teori til empiri» og induktiv handler om å gå fra «empiri til teori», eller som Johannessen et al. (2011) kaller det, «å trekke slutninger fra det generelle til det konkrete» og vice versa. I denne studien ble ingen av tilnærmingene sett på som *perfekte*. I stedet ble det valgt en tilnærming som ligner det Tjora (2017) kaller SDI, stegvis-deduktiv induktiv metode. Med SDI som metode beveger man seg frem og tilbake mellom rådata og konsepter eller teorier (Tjora, 2017). Denne tilnærmingen ble valgt fordi vi i utgangspunktet ønsket en induktiv tilnærming, men hadde for lite kunnskap om temaet til å kunne utføre forskningen rent induktivt. Å si at vi har brukt SDI systematisk som Tjora (2017) presenterer det, er langt på vei en overdrivelse, men vi har i noen grad beveget seg frem og tilbake mellom

en induktiv og en deduktiv tilnærming i denne studien. Studien bærer mest preg av en induktiv tilnærming, noe som vil fremkomme i de påfølgende delkapitlene.

3.3.1 Case

Formålet med denne oppgaven var å se nærmere på hvordan to delvis adskilte organisasjoner tilnærmer seg bruken og tilpasning av agile metodikker. Dette vil si at vi ønsket å opparbeide oss dybdekunnskap om et eller flere fenomen i to ulike situasjoner og se på likheter og forskjeller i de to situasjonene. Med to delvis adskilte organisasjoner mener vi at Telenor Norge og Telenor Digital teknisk sett er en del av samme konsern, men opererer i stor grad adskilt fra hverandre. For enkelhetens skyld vil de i denne oppgaven omtales som to adskilte organisasjoner.

Med utgangspunkt i dette ble casestudie ansett som mest hensiktsmessig forskningsdesign. I denne studien skal vi prøve å besvare spørsmål som «hvilken», «hvordan» og «hvorfor» for flere kontekster, og ifølge Yin (2009) er da casestudie et hensiktsmessig design. Denne studien benytter seg av et flercasedesign, da vi skal se på både Telenor Digital og Telenor Norge. For å gjennomføre dette har vi valgt å basere oss på flere analyseenheter som får frem ulike perspektiv i form av 8 dybdeintervju med prosjektledere, avdelingsledere og utviklere, samt 2 ustrukturerte intervju med avdelingsledere. Dette kjennetegner det Yin (2009) omtaler som en analytisk tilnærming.

3.4 Datainnsamling og analyse

I denne studien ble data samlet inn gjennom intervjuer av personer ansatt i Telenor Norge og Telenor Digital. I dette delkapittelet kommer prosessen med å samle inn og analysere data til å bli nærmere beskrevet. Dette inkluderer bakgrunn for valg av datainnsamlingsmetode, hvordan datainnsamlingen ble gjennomført, bakgrunn for valg av datakilder, litteratursøk og beskrivelse av hvordan dataanalysen ble gjennomført.

3.4.1 Intervju

I denne studien hadde vi som formål å undersøke et fenomen i to kontekster. Vi hadde ikke som mål å generalisere funnene til resten av bransjen, og det ble derfor vurdert som mest relevant å se på fenomenet i dybden. Med dette mener vi at vi ville gå inn på nyanser og spesifikke hendelser i de to kontekstene. For å gjøre dette ble dybdeintervju vurdert for å være den mest hensiktsmessige datainnsamlingsmetoden. Det var også et ønske om å gi intervjuobjektene mest

mulig frihet til å snakke om det de så på som interessant, uten at det gikk helt utenfor temaene i forskningsopplegget. For å oppnå dette vurderte vi semistrukturerte intervju som den mest relevante fremgangsmåten. Ifølge Johannessen et al. (2011) er semistrukturerte intervju den mest brukte måten å samle inn data på i kvalitative studier, og hensiktsmessig hvis man trenger å gi intervjuobjektene større frihet enn hva et spørreskjema eller et strukturert intervju tillater. Den semistrukturerte tilnærmingen mener vi bidro til at studien fikk en del data som ikke var relevant, i stor grad fordi vi tillot intervjuobjektene å snakke veldig fritt. Likevel ga det oss også verdifull data vi ikke hadde regnet med i forkant, da friheten ga oss mulighet til å stille improviserte oppfølgingsspørsmål underveis. Intervjuene ble dermed mer åpen enn vi hadde forventet, en faktor vi mener betraktes både som en styrke og en svakhet.

På bakgrunn av det innledende litteratursøket samt innledende intervju ble det avdekket en rekke tema som ble ansett som interessante. I forkant av intervjuene ble intervjuobjektene tilsendt litt informasjon om hvilke overordnende tema som ville være hovedfokus, samt et par enkle spørsmål om hvem de var og hvilken erfaring de hadde med de ulike temaene. Dette ble gjort for å være best mulig forberedt til intervju, samt at intervjuobjektene skulle gjøre seg noen tanker om tema som skulle tas opp. Intervjuguide var basert på innledende intervju, litteratursøk samt svarene fra intervjuobjektene.

3.4.2 Intervjuguide og gjennomføring av intervju

Intervjuene baserte seg på en semistrukturert tilnærming som beskrevet i Johannessen et al. (2011). Det ble laget en intervjuguide som er delt inn i innledning, faktaspørsmål, introduksjonsspørsmål, hovedspørsmål og avslutning, se vedlegg 1. I innledningen ble det presentert informasjon om prosjektet, databehandling o.l. Dette ble gjort for at intervjuobjektene skulle føle seg trygge, samt å tydeliggjøre hvilke tema vi anså som interessante. Videre ble enkle faktaspørsmål som «hvor lenge har du jobbet i Telenor?» stilt. Noen av disse spørsmålene ga data som kunne være nyttig, men i utgangspunktet ble spørsmålene stilt for å få en rolig start på samtalen og bli bedre kjent med intervjuobjektene. Videre ble intervjuobjektene stilt et par generelle introduksjonsspørsmål. Disse ble stilt på formen «Hva tenker du når du hører agil arbeidsmetodikk?». Disse spørsmålene ga verdifulle svar om hvilke generelle tanker intervjuobjektene hadde rundt agil arbeidsmetodikk. Intervjuobjektene ble spurt om å ikke trekke inn Telenors spesifikke arbeidsmetodikk i disse spørsmålene. I Hovedspørsmålene gikk vi nærmere inn på hvordan intervjuobjektene så på deres egen arbeidsmetodikk. Her ble det, så langt det lot seg gjøre, stilt åpne spørsmål om agil

arbeidsmetodikk i Telenor. Forskerne ønsket å holde spørsmålene så åpne som mulig slik at intervjuobjektet fikk dra frem det han/hun anså som viktig. Vi stilte så oppfølgingsspørsmål om det som ble ansett som relevant. I intervjuguiden hadde vi forberedt enkelte oppfølgingsspørsmål til hvert hovedspørsmål, men mange av oppfølgingsspørsmålene kom vi på underveis. I hoveddelen fokuserte vi på å få intervjuobjektene til å gi eksempler så ofte som mulig. Helt til slutt fikk intervjuobjektene mulighet til å ta opp andre ting som de følte ikke ble dekket tidligere i intervjuet.

Vi valgte å fordele arbeidet i dybdeintervjuene slik at en av oss hadde hovedansvar for å gjennomføre intervjuet, mens den andre tok notater og hadde hovedansvar for å stille oppfølgingsspørsmål. Intervjuene ble i tillegg tatt opp, slik at vi kunne ha fokus på selve intervjuet og få med alle detaljer som ble nevnt. Intervjuobjektene ble muntlig informert om opptak og at dette var valgfritt, samt at de når som helst kunne avbryte intervjuet uten å måtte oppgi grunn. Intervjuobjektene ble videre informert om at intervjuene ville bli transkribert og anonymisert, og at etter prosjektet slutt ville all rådata slettes. Tiden avsatt til intervjuet var i utgangspunktet en time, men ble noen ganger lengre og noen ganger kortere avhengig av hva intervjuobjektene hadde å si. Lengden på intervjuene gikk i fra 50 min på det korteste til 75 min på det lengste. Hovedparten av intervjuene ble gjennomført ansikt til ansikt, mens to intervju ble gjennomført over telefon av praktiske årsaker.

3.4.3 Utvalg av informanter

Når utvalg av intervjuobjekt til dybdeintervju skulle gjøres, ble det lagt vekt på vekt på 3 ting, (1) erfaring med bruk av agile metodikker, (2) mangfold mellom ledere og utviklere, og (3) mangfold mellom intervjuobjekt fra Telenor Norge og Telenor Digital. Dette er hva Johannessen et al. (2011) kaller strategisk utvalg, noe som er vanlig i kvalitativ forskning. Bakgrunnen for kriteriene var at vi hadde en hypotese om at dette ville skape rikest mulig empiri samtidig som det var gjennomførbart. Av totalt åtte intervjuobjekt som ble valgt ut til dybdeintervju jobbet henholdsvis fem i Telenor Norge og tre i Telenor Digital. Tre av intervjuobjektene i Telenor Norge hadde ledelse som hovedoppgave, mens ett av intervjuobjektene i Telenor Digital var primært leder. Resten av intervjuobjektene var i hovedsak utviklere, og alle hadde erfaring med agile metodikker for programvareutvikling. For å sikre at kriteriene ble oppfylt før intervjuet, ble dette spurt om i spørsmålene som ble tilsendt intervjuobjektene. Enkelte av intervjuobjektene ble også anbefalt av andre intervjuobjekt, da de mente at de kunne ha mye å si om temaet. Vi mener dette utvalget var tilstrekkelig til å svare

på problemstillingen, da det var bredt nok til at vi fikk belyst relevante aspekt fra både leder- og utviklerperspektiv, samt at vi fikk belyst fenomenet tilstrekkelig både i Telenor Norge og i Telenor Digital. Det var nødvendig å ha flere intervjuobjekter fra Telenor Norge, da svarene fra intervjuobjektene spriket i større grad enn hva de gjorde i Telenor Digital, noe som sannsynligvis kommer av at Telenor Norge er en større og mer diversifisert bedrift, noe som gjorde at intervjuobjektene hadde mer ulik erfaring med arbeidsmetodikk i bedriften enn i Telenor Digital.

3.4.4 Litteratursøk

Som tidligere nevnt gjennomførte vi et innledende litteratursøk før gjennomføring av dybdeintervjuene. Søkemotorene Oria og Google Scholar ble hovedsakelig brukt for å gjøre litteratursøk, hvor vi innledningsvis søkte på fraser som «agile», «Method tailoring» og «project management». I tillegg ble det benyttet tidligere masteroppgaver og litteraturstudier for å finne relevante kilder. Bakgrunnen for litteratursøket var at vi hadde avdekket et par tema gjennom de innledende intervjuene som vi trodde kunne være interessante, men vi ikke hadde tilstrekkelig kunnskap om. Ved å gjennomføre et innledende litteratursøk fikk vi mer kunnskap om hvilken forskning som eksisterte på fagområde, og om det var noen spesifikke tema som var mer interessante enn andre. På bakgrunn av dette ble tilpasning og bruk av agile metodikker valgt som tema, med fokus på forskjeller i små og store bedrifter.

Etter at dybdeintervjuene var gjennomført, gjorde vi et nytt litteratursøk, denne gangen med mer spesifikke fraser rettet mot funn i oppgaven som usikkerhet, retrospektiv og formalitet. Vi fikk også verdifull hjelp fra ekstern veileder som har betydelig erfaring fra feltet. Det er verdt å nevne at enkelte av funnene våre er lite forsket på tidligere, noe som har begrenset tilgangen på relevant teori.

3.4.5 Dataanalyse

For å kunne analysere den innsamlede dataen, måtte den først struktureres. De transkriberte intervjuene ble derfor kodet med analyseprogrammet NVivo 11. Dataen ble kodet i forskjellige kategorier, eller «koder», som ble ansett som relevante for videre analyse. Formålet med analysen var å trekke ut mønstre som vi senere kunne knytte opp mot relevant teori. Kodene ble i hovedsak basert på interessante tema fra datamaterialet, fremfor teori eller problemstilling. Dette i tråd med Tjora (2017), som sier at ved bruk av SDI som forskningsmetode skal kodene være basert på datamaterialet. Siden et innledende litteratursøk ble gjennomført før

datainnsamlingen, var nok noen av kodene delvis basert på teori. Dette er også i tråd med hans påstand om at ren induktiv tilnærming med «tomt hode» er umulig i praksis, og at man alltid vil være påvirket av teori i analysearbeidet. Det er også verdt å påpeke at siden intervjuguiden var påvirket av det innledende teorisøket, var også spørsmålene og dermed også svarene til intervjuobjektene påvirket av dette. Etter at dataen var ferdig kodet, gikk forskerne igjennom hovedkategoriene for å se hvilke funn som var mest interessante, og disse ble videre grunnlag for resultat og diskusjonskapittelet i studien.

3.5 Forskningsoppleggets kvalitet

I dette delkapittelet skal vi se nærmere på kvaliteten til forskningsopplegget. For å gjøre dette kommer relabiliteten og validiteten til forskningsopplegget til å bli nærmere beskrevet.

3.5.1 Relabilitet

Relabilitet, eller pålitelighet, sier noe om hvor pålitelig funnene i en studie er. Relabilitet omhandler hvor nøyaktig dataen er, hvilken data som er benyttet, hvordan dataen er innsamlet og hvordan dataen er analysert (Johannessen et al., 2011). I kvantitative studier snakker man ofte om relabilitet i form av repliserbarhet. Dette er noe som ikke er av like stor relevans i kvalitative studier, da det i praksis er umulig å replisere kvalitative studier (Johannessen et al., 2011). I kvalitative studier er det i stedet snakk om hvilke faktorer som kan ha vært med å forme resultatet av studien, og hva forskerne eventuelt har gjort for å sikre best mulig relabilitet. I denne studien er det derfor mer hensiktsmessig å trekke frem forskningens pålitelighet, da pålitelighet i dette tilfellet handler om å gi leseren dokumentasjon på hvordan ulike tolkninger og avgjørelser har blitt tatt (G. Guba og S. Lincoln, 2018). For å unngå påvirkning har vi vært opptatt av å la egne tanker om agile metodikker påvirke svarene til intervjuobjektene minst mulig. For å få til dette ble intervjuobjektene i hovedsak stilt åpne spørsmål. Lukkede spørsmål ble kun stilt som oppfølgingsspørsmål hvis det var noe spesifikt vi var ute etter å avdekke, og når det var tilfellet forsøkte vi å unngå at spørsmålene var ledende. For å få mest mulig nøyaktig data, ble alle intervjuene transkribert så nøyaktig som mulig. Det vil si at pauser, nøling og lignende ble inkludert i transkripsjonen. Dette gjorde at vi kunne sikre at det var intervjuobjektenes faktiske mening som kom fram. I resultatkapittelet er derimot sitater delvis renskrevet for å oppnå bedre lesbarhet. Ved renskrevet menes at pauser, nøling og ord som ikke ble sett på som betydningsfull for meningen bak er tatt bort. Ord som «på en måte», «slik», «liksom» og lignende er inkludert da vi har ansett det som betydningsfullt. Disse valgene kan

ha påvirket fremtoningen av sitatene, men vi har prøvd å sikre den faktiske meningen, og dermed relabiliteten, så godt som praktisk mulig. Alle intervjuobjektene ble også tilbydd å få ferdig transkripsjon tilsendt etter intervjuet for gjennomsyn og eventuell korreksjon av meninger. Ved å vise åpenhet rundt forskningsprosessen mener vi fremgangsmetoden har vært med på å styrke påliteligheten til studien og gitt innsikt i hvordan konklusjonen har blitt til.

3.5.2 Validitet

Ifølge Johannessen et al. (2011) omhandler validitet hvor godt dataene faktisk representerer virkeligheten. Spørsmålet «måler vi det vi tror vi måler?» er sentralt når det kommer til validitet. Johannessen et al. (2011) skiller mellom tre typer validitet: begrepsvaliditet, intern validitet og ekstern validitet.

Begrepsvaliditet sier noe om forholdet mellom fenomenet som skal undersøkes, i dette tilfellet tilnærmingen til bruk og tilpasning av agile metodikker i Telenor Norge og Telenor Digital, og dataen som presenteres, i dette tilfellet data fra dybdeintervju. For å sikre begrepsvaliditeten har vi vært nøye med å presisere hva som menes med tilnærming gjennom forskningsspørsmål og videre avgrensinger, se kapittel 1.6. Studien er også utarbeidet av to forskere som videre styrker begrepsvaliditeten. I tillegg har vi gjennom hele studien fått kontinuerlig veiledning av både to interne veiledere, samt en ekstern veileder med agile metodikker som fagområde.

Intern validitet dreier seg om hvorvidt en studie er egnet til å påvirke årsakssammenhenger eller ikke (Johannessen et al., 2011). Ifølge Yin (2009) vil en studies interne validitet styrkes hvis funnene i empirien matcher funn fra tidligere forskning. I denne studien ble den interne validiteten styrket ved å prøve å gjøre nettopp dette. I tillegg ble det lagt vekt på å ikke påvirke intervjuobjektene med egne tanker om agile metodikker, noe som ville ha svekket den interne validiteten (Johannessen et al., 2011). Det er viktig å poengtere at konklusjon om eventuelle årsakssammenhenger i denne studien naturligvis vil være subjektive og påvirket av vår egen oppfatning av dataen, noe som kan ha svekket den interne validiteten. Det kan også stilles spørsmål på om vi har klart å unngå å påvirke intervjuobjektene med egne meninger om agile metodikker, selv om hensyn ble tatt.

Ekstern validitet omhandler om studiens resultater kan generaliseres eller overføres til andre settinger enn det som er undersøkt (Johannessen et al., 2011). I denne studien er det uklart om funnene kan generaliseres utenfor de gitte casene. Det er også viktig å nevne at Telenor Norge er en stor organisasjon, så det er ikke sikkert at alle funnene kan generaliseres til *hele* Telenor

Norge. Hensikten med denne studien var ikke å finne funn som kunne generaliseres til en hvilken som helst populasjon, men heller å øke innsikten om hvordan tilnærminger Telenor Norge og Telenor Digital har til agile metodikker. Skulle man generalisere ville det vært hensiktsmessig med en større og mer representativ populasjon.

3.6 Etiske vurderinger

Forskning må, som all annen virksomhet, forholde seg til etiske retningslinjer og regler (Johannessen et al., 2011). Denne studien dreier seg i stor grad om enkeltpersoners meninger og synspunkt, noe som fordrer at vi som forskere tar etiske hensyn for forskningsopplegget vårt. Intervjuobjektene ble alle garantert anonymitet i forkant av intervju. Transkripsjoner ble anonymisert, samt at vi garanterte at all data som kan spores tilbake til enkeltpersoner skal slettes etter prosjektets slutt. Intervjuobjektene ble også tilbydd å få transkripsjon tilsendt for oversyn og eventuell presisering av meningen og synspunkt. Siden personopplysninger som navn ble samlet inn, ble det søkt om godkjennelse av Norsk Samfunnsvitenskapelig Datatjeneste (NSD) om å gjennomføre datainnsamling, i tråd med gjeldende lover og regler. Dette ble godkjent før gjennomføring av første intervju, se vedlegg 2. Vi vurderer derfor at etiske hensyn er ivaretatt, da alle intervjuobjektene samtykket til å delta, personopplysninger er behandlet på fortrolig måte og konsekvenser ved å delta i studien anses som svært lave.

3.7 Forskningsoppleggets begrensninger

Denne studien har flere begrensninger. En begrensning har vært at flesteparten av intervjuobjektene fra Telenor Norge holder til i Oslo, noe som har gjort at enkelte intervju har blitt gjennomført over Skype/telefon av praktiske årsaker. Dette kan ha ført til at forskerne gått glipp av, eller mistolket informasjon fra intervjuobjektene. Det at intervjuobjektene har fått muligheten til å se over transkripsjonen for å sikre at meningsinnholdet kommer korrekt frem, har redusert denne risikoen noe.

Det er også viktig å nevne at mange av funnene i denne studien dreier som om tilpasning av arbeidsmetodikker. I denne studien har dette fordret at intervjuobjektene har reflektert over tidligere erfaringer. Hadde denne studien vært gjennomført over en lengre periode kunne kanskje forskerne sett utviklingen av arbeidsmetodikk på en mer nøyaktig måte. Dette var dessverre ikke mulig i denne oppgaven, men er noe forskerne anbefaler for mulig videre forskning.

Det at vi hadde relativt lite kunnskap om fagfeltet i forkant av studien er også noe som kan ses på som en svakhet, da det tidvis har vært vanskelig å vite hva som burde være fokusområde. Dette er for så vidt noe forskerne ser på som en styrke, da et «ferskt» syn på feltet kan bidra til interessant refleksjon og forhåpentligvis gode, nye funn. Veiledere har her vært til stor hjelp med å peke oss i riktig retning, samt bidra med forslag til relevant forskning å ta utgangspunkt i. Det er også verdt å poengtere at enkelte av fokusområdene i denne studien ikke har blitt forsket mye på tidligere. Det er for så vidt dette som gjør forskningen interessant, men gjør det også vanskelig å støtte opp funnene med relevant forskning.

4 Resultat

I dette kapittelet kommer utvalgte empiriske funn fra datainnsamlingen til å bli presentert. Dataene ble som nevnt i kapittel 3.4.5 kodet i kategorier som ble ansett som relevant for å besvare på problemstillingen, og det er de kodede dataene som vil bli presentert i dette kapittelet. Dette kapittelet vil først presentere de agile arbeidsmetodikkene som benyttes i henholdsvis Telenor Digital og Telenor Norge, før funn relatert til hvordan Telenor Digital og Telenor Norge tilpasser ulike agile metodikker til sitt bruk blir presentert til slutt. Alle funn i dette kapittelet er hentet fra dybdeintervjuene som gjennomført i forbindelse med denne studien, se kapittel 3.4.1 for nærmere beskrivelse.

4.1 Agil arbeidsmetodikk Telenor Digital

For å finne ut av hvordan Telenor Digital jobber agilt, ble alle intervjuobjektene innledningsvis bedt om å forklare hva de mener kjennetegner agil arbeidsmetodikk. Videre ble de bedt om å forklare hvordan de jobber med programvareutvikling i teamet og om de eventuelt benyttet noen agile praksiser. Det ble også stilt spørsmål rundt hvordan de er organisert, teamstørrelser og lignende.

Intervjuobjektene trekker i stor grad fram det samme på spørsmål om hva det vil si å jobbe agilt på generell basis. Alle intervjuobjektene nevner at det å jobbe i iterasjoner, og å dele opp oppgaver er helt sentralt. Dette er ifølge intervjuobjektene viktig for å kunne jobbe og levere oppgaver hurtigst mulig. Det å få mulighet til å teste ut programvare underveis er også noe som intervjuobjektene anser som viktig.

> *«Smidig arbeidsmetodikk, det er mange elementer i det, men det er å ha en metodikk hvor man jobber med å bryte ned problemer i mindre problemstillinger slik at du kan gjennomføre forbedringer hurtig og få testet ut ting. For meg så er det i hvert fall en sterk knytning mellom stegvis forbedring og smidig metodikk.»*

Fleksibilitet og evne til å skifte fokus hurtig er også noe av det intervjuobjektene trekker frem som sentralt med agil arbeidsmetodikk.

> *«Fleksibelt, det er vel det ordet egentlig betyr, men jeg relaterer det veldig til Scrum. At du har sprinter, at man kan skifte fokus innen kort tid... Føles veldig fleksibelt, at man kan takle at prioriteringa endrer seg.»*

4.1.1 Hvordan jobber Telenor Digital?

Telenor Digital består ifølge intervjuobjektene av to team pr i dag. Et frontend-team, og et backend-team. Dvs. at teamene er delt inn etter hvilken del av systemet de utvikler for. Størrelsen på teamene varierer til tider, men ligger ifølge intervjuobjektene på mellom 8 og 12 stk. pr team. Det har også vært eksperimentert med et stort team i stedet for to, men dette fungerte dårlig. Teamene har ulike arbeidsoppgaver, men jobber stort sett etter samme arbeidsmetodikk, med noen ulikheter som vi kommer tilbake til. Når intervjuobjektene blir bedt om å beskrive arbeidsmetodikken de jobber etter i Telenor Digital, trekker alle frem sprintene som sentrale. Ifølge intervjuobjektene jobber de sprintbasert med 2-ukers sprinter, eller bolker, der oppgavene som skal gjøres har ett eller få temaer slik at det blir enklere å holde fokus. Grunnen for at de bruke 2-ukers sprinter er fordi at det er en del møter i overgangen mellom to sprinter som tar tid, og ved 1-ukers sprinter tar dette for mye av sprinten. På den andre siden mener de at 4 uker blir for lang tid per sprint.

> *«…4 blir for langt igjen. Helt umulig å planlegge for 4 uker. 2 er sånn ganske greit. 3 kunne kanskje også ha gått men 2 er det vi har sagt at vi holder på med akkurat nå.»*

Hvilke oppgaver som skal utføres av teamet i hver sprint, er det avdelingsleder og en product manager (PM) som har ansvar for å bestemme. Dette blir bestemt på bakgrunn av ønsker fra forskjellige interessenter, teamets ønsker, kapasitet og avtalte leveranser. Før hver sprint har de et møte kalt «sprint planning» hvor de planlegger hvilke oppgaver som skal gjøres i neste sprint. Her har også teamet mulighet til å komme med innspill på hva som burde gjøres. Dette er ifølge intervjuobjektene viktig fordi de har såpass dyp teknologisk innsikt at mener de vet best hva som er riktig å gjøre.

> *«Jeg tror her så føler vi et større ansvar for å bestemme hva som skal gjøres, vi har såpass dyp teknisk innsikt i plattformen vår og vet hva som er riktig, det som kalles technology push. Vi har ting vi gjerne ville ha gjort, som kanskje ingen stakeholders har som punkter på backloggen fordi de vet ikke at de vil ha det eller det er ikke viktig for dem.»*

Oppgaver som kan være aktuelle å ha med i sprinten kommer inn kontinuerlig, og organiseres i en backlog som er en slags to-do liste. Avdelingsleder og PM oppdaterer og revurderer hele tiden denne for å ha den i best mulig stand til en sprint starter. Hvis det blir bestemt at en oppgave skal gjøres i en sprint, tas den ut av backloggen og «inn» i sprinten. Hvis oppgaven ikke blir fullført i løpet av sprinten blir den enten satt tilbake til backloggen, tatt med i neste

sprint eller forkastet. Etter hver sprint har de også et møte kalt «sprint review». Sprint review er et møte de holder etter hver sprint hvor de går igjennom hva som er gjort og hva som ikke ble ferdig. Her er ofte interessenter også med, slik at de får en oppdatering og kan komme med innspill til hva de ønsker.

Avdelingsleder får inn mulige aktuelle oppgaver fra i hovedsak tre forskjellige kilder:

1. Kunder som vil ha en endring i produktet Telenor Digital leverer til dem.
2. Personer innad i teamet som ønsker å gjøre en endring.
3. Feil i eksisterende kode (bugs).

I Telenor Digital trekker intervjuobjektene frem sprinter og oppdeling av arbeidsoppgaver som en sentral del i hvordan de arbeider, noe som trekkes frem som en stor faktor til at de evner å skifte fokus og arbeide fleksibelt. Ved å bli inkludert i planleggingsprosessen med interessenter opplever intervjuobjektene et stort ansvar. Dette mener intervjuobjektene forbedrer estimering og legger til rette for bedre løsninger da teamet har innsikt i forbedringspotensialet.

4.1.2 Agile praksiser

Før hver sprint har de også et møte kalt «refinement». I dette møte skal avdelingsleder, PM og teamet estimere hvor store de forskjellige arbeidsoppgavene er i forhold til hverandre. Hver oppgave blir tildelt en relativ størrelse, kalt «storypoints», som er enten 1, 3, 5, 8 eller 13, hvor 1 er en veldig liten oppgave, og 13 er så stor at den må deles opp i mindre oppgaver. Poenget med refinements er å få estimert oppgaver slik at sprint planning blir best mulig. Intervjuobjektene trekker også frem at de sliter med å gjennomføre de større oppgavene, og at de må brytes ned mer.

> *«…vi sliter med 8ere, de er for store. Vi er nødt til å bryte ned 8erne, veldig sjelden at vi klarer å kjøre 8erne på en sprint selv om vi tror at vi klarer det»*

Estimering er utfordrende da oppgavens kompleksitet øker i takt med størrelsen, dette fører ofte til at deler av oppgaven blir stående utenfor sprinten og må tas med i neste sprint. Ifølge intervjuobjektene blir det også vanskeligere å dele opp oppgavene etter hvert som de blir mer komplekse da de har avhengigheter med hverandre. De trekker også frem at de ofte undervurderer vanskeligheten av oppgavene, og at det fører til at det hoper seg opp med oppgaver som ikke blir ferdigstilt.

«... med alle sånne IT prosjekter så har du en eksponential kurve i kompleksitet etter som størrelsene vokser, og vi klarer ikke når det blir for stort å ta høyde for hvor vanskelig det egentlig er, det dukker opp nesten alltid opp på sånne store prosjekt. Og da drar det ut i tid og plutselig så får du ikke gjort ferdig sprinten, så står du igjen med en liten rest til neste sprint, også har du det gående.»

Intervjuobjektene trekker frem at det er viktig å gjøre en god jobb i refinement, slik at neste sprint blir enklere å utføre. Det poengteres også at refinements er noe som teamet gjør, ikke bare ledelsen.

«… Å bryte de ned og beskrive de tydeligere sånn at de er så veldefinerte som mulig, det er en oppgave som skjer med teamet.»

Det at teamene deltar aktivt på planleggingen og estimeringen trekker intervjuobjektene frem som kjempeviktig, da det gjør at teamene føler seg mer forpliktet til å levere det man har planlagt i sprinten.

Etter hver sprint er ferdig, har teamet et møte de kaller retrospektiv. I disse møtene er det PM, teamet og eventuelt avdelingsleder som deltar. Poenget med møtet er ifølge intervjuobjektene å snakke om hvordan man jobbet i sprinten som var, og om det er noe man kan gjøre annerledes i neste sprint. Nøyaktig hvordan retrospektivene gjennomføres forklares best av intervjuobjektene selv:

«I retrospektive så sitter vi først og skriver på Post-its, så skriver du en ting på hver lapp, så får vi kanskje 5 eller 3 minutter til å gjøre det, det kommer an på hvor mye man har å si. Så tar vi etter tur å forklare litt om det vi har skrevet, bare sånn kjapt, det kan være en setning eller det kan være at man trenger litt mer tid. Så limer vi det på veggen, også prøver vi å samle når alle er ferdig så grupperer vi det slik at det er litt samme tema … så ser vi dem som har fått mest kommentarer eller som ser viktigst ut, så diskuterer vi det litt også kommer vi frem til en enighet.»

Det at man skriver ned ting på Post-its er ifølge intervjuobjektene en god måte å få med alles mening på, også de som ikke nødvendigvis snakker høyest i møter ellers.

I selve sprinten benytter de seg av daglige 15-minutters standups gjennom hele sprinten. I standupene går man ifølge intervjuobjektene gjennom oppgaver og sier hva man jobber med, om man er ferdige eller om man har noen problemer man trenger innspill til. Dette er noe

enkelte av intervjuobjektene så på som til tider unødvendig, men det ble også poengtert at Stand-up var en fin måte å få innspill fra andre hvis man hadde problemer.

I det ene av de to teamene har de også tatt etablert en rolle de kaller for «sweeper». En sweeper er en rolle som en eller flere på teamet har, og den person jobber i praksis utenom sprinten. Sweeperen skal ta seg av oppgaver som kommer inn mens sprinten er i gang slik at man skal få gjennomført hasteoppgaver uten at hele teamet mister fokus på oppgavene i sprinten. Sweeperen bruker et «Kanban-brett» som et verktøy for å organisere og optimere arbeidsflyten. Et av intervjuobjektene forklarer rollen slik:

> *«… så ser vi at vi har veldig mange flater å jobbe mot, vi er integrert i 50 ulike tjenester. Det betyr at fra disse 50 kommer det til enhver tid mange ulike forespørsler, det kan være spørsmål, bugs, endringsønsker. Noen av disse haster ganske mye, og hvis vi har planlagt en sprint, så må man hele tiden gjøre avvik. Så en av de tingene vi har gjort er det vi har definert som en sweeper, en arbeidsoppgave som i utgangspunktet går på rundgang selv om det som regel bruker å være de samme personene, som ikke følger sprinten. De tar løpende oppgaver, og kommer det ikke løpende oppgaver så prøver de å plukke småoppgaver som i utgangspunktet ikke er definert inn i sprinten. Så vi prøver på backlogen å ha noen oppgaver som er lette å plukke, slik at man har litt tid til overs.»*

Dette er altså en rolle de har valgt å bruke på det ene teamet, men ikke det andre. Ifølge et av intervjuobjektene har de ikke tatt det i bruk fordi de mener det ødelegger litt for forutsigbarheten i sprinten.

> *«Vi har ikke tatt det i bruk, vi syns det virket litt merkelig. Fordi at i sprinten har du 2 uker og så mange storypoints som du skal gjøre ferdig, og det er forutsigbart – det skal være forutsigbart, men om du i tillegg har en Kanban-board så blir det på en måte å jukse til seg ting man kan ta inn likevel. Fordi du vet ikke hvor mye som kommer inn så du vet ikke hvor mye sweeperen kan bidra på sprinten.»*

Videre har denne rollen utviklet seg fra å være bare en enkelt person, til å bli et team på 4. Dette er ifølge intervjuobjektene på grunn av at de ser at de uforutsette oppgavene som må gjennomføres kontinuerlig ble for store for bare en person. Det nye teamet kaller de for et «live-team», som altså er 4 personer som tar unna uforutsette oppgaver som ikke er relatert til sprinten. Dette er ifølge intervjuobjektene et resultat av at oppgavene blir større, og at det blir for lite med en person til å ta seg av dem. Praksisen er noe som endres kontinuerlig, og ved

intervjutidspunkt besto live-teamet av totalt 4 personer. Live-teamet tar ikke bare uforutsette oppgaver, men også oppgaver av en viss størrelse som egentlig ikke passer inn i sprinten.

> *«Tidligere så hadde vi en person som «eide» et Kanban-board, det var veldig små tasks og det var han som egna seg som en sweeper. Men det vi har sett nå at vi har noen oppgaver som ikke er helt tematisk riktig å ha i sprinten men som er av viss størrelse sånn at det ikke egentlig fungerer å bare ha en person til å jobbe med det. Akkurat nå har vi 4 personer som jobber i det Kanban-boardet, det har vi valgt å kalle for live det teamet».*

Intervjuobjektene poengterer at live-teamet ikke bare er nyttig for å ta unna oppgaver som haster, men også generelt for at man ikke skal gå «tom» for oppgaver. Teamet ble opprettet delvis som en konsekvens av at utviklere gikk tomme for oppgaver på selve sprinten, og begynte å plukke andre oppgaver fra backloggen. Det at man hele tiden har arbeidsoppgaver er ifølge intervjuobjektene viktig for å føle at man er effektive.

> *«Så det er rett slett at folk føler at de til enhver tid føler at de har en action (oppgave) de kan plukke som er nyttig at de tar i stedet for at de føler at de ikke får brukt tiden sin. Det var min primære motivasjon for å prøve å gjøre det sånn.»*

Teamet i Telenor Digital opplever refinements som en viktig praksis for å kunne estimere hvilke ressurser som kreves for de ulike oppgavene. Et av intervjuobjektene trekker frem eierskap som en av fordelene de oppnår ved å inkludere teamet i planleggingen, da teamet selv setter fristen og dermed forplikter til å fullføre oppgaven innen tidsfrist uten forsinkelser. I etterkant av sprinten trekker intervjuobjektene frem at retrospektiv fungerer som en nyttig praksis for fremtidige sprinter. Dette ser vi kan være hensiktsmessig med tanke på tilpasning da Telenor Digital får gjort små endringer kontinuerlig og blir mer effektiv til neste sprint. Dette mener et av intervjuobjektene også gjør det mulig å være mer effektiv ved at de unngår sløsing av ubrukte ressurser.

4.1.3 Oppsummering

Telenor Digital har en arbeidsmetodikk som minner mye om Scrum. De jobber i 2-ukers sprinter, med planning og refinement før hver sprint, og review og retrospektiv etter hver. De er organisert i 2 team som jobber ganske likt, men med den ene forskjellen at det ene teamet har implementert noe kalt et live-team som tar seg av oppgaver som må gjøres med en gang. Dette gjør at sprinten kan fullføres uten forstyrrelser og teamet får håndtert uforutsette hendelser

som krever umiddelbar oppmerksomhet. Refinements og retrospektiver sørger for at teamet blir inkludert i planlegging og forbedring slik at programvareutviklingen kontinuerlig blir forbedret.

4.2 Agil arbeidsmetodikk Telenor Norge

Organiseringen i Telenor Norge er mer kompleks enn i Telenor Digital. Der hvor Telenor Digital i praksis består av 2 team i 1 avdeling, består Telenor Norge av flere ti-talls team i mange avdelinger. I noen av avdelingene jobbes det primært agilt, i andre primært etter fossefallsmetodikk og noen avdelinger benyttes en slags kombinasjon. Dette avhenger av arbeidsoppgaver, størrelse på team og avhengigheter opp mot andre team. Dette er noe av det vi skal se nærmere på i dette delkapittelet.

På spørsmål om hva det innebærer å jobbe agilt, svarte flere av intervjuobjektene at agilitet er evnen til å endre seg underveis og respondere på uforutsette ting. Også i Telenor Norge trekkes det frem at det å dele opp oppgaver er en sentral del av det å jobbe agilt.

> *«Agilitet er evnen til å endre seg underveis, ikke sant? Respondere på ting du ikke kunne forutsett på forhånd. Så da bryter du ned oppgaver i sprinter, og har retrospektiv, og så ser du om vi skal endre litt retning og sånn».*

Et av intervjuobjektene trekker også frem det at IT-bransjen historisk har hatt en overdreven evne til å forutse fremtiden, og at man derfor har behov for agile metodikker som legger mer til rette for å respondere på endringer.

> *«… jeg tror at man tidligere innenfor IT har hatt en overdreven tro på evne til å forutse fremtiden.»*

De andre intervjuobjektene trekker også frem mye av det samme, hvor det å ikke bli låst til en løsning tidlig i prosessen er noe de ser på som viktig, og at det å jobbe agilt i stor grad handler om å levere kontinuerlig i stedet for at komplette produkt blir levert etter 2 år med utvikling.

Det er i stor grad samme oppfatning om at å jobbe agilt innebærer å kunne endre seg underveis ved å respondere på uforutsette hendelser og dele opp arbeidet i mindre biter for å levere kontinuerlig, og på den måten håndtere usikkerhet i prosjektet.

4.2.1 Hvordan jobber Telenor Norge?

Telenor Norge består av mange avdelinger som utvikler, retter og forbedrer programvare. Ifølge prosjektstyringsgruppen er det et skille mellom de som arbeider med kritiske tjenester knyttet

opp mot mobilnettet og personvern, og de som arbeider med ikke-kritiske tjenester som applikasjoner, nettløsninger og lignende. Dette skillet avgjør i stor grad hvilken arbeidsmetodikk som blir benyttet.

Telenor har tidligere utelukkende vært en tilbyder av telefontjenester som sørget for å forvalte telenettverket og sikre at Norges innbyggere kunne kommunisere. De ansatte har derfor ifølge intervjuobjektene et stort ansvar med å bygge og drifte en samfunnskritisk tjeneste som må være stabil og sikker:

> *«Vi hadde bare en tjeneste, og folk var bare happy hvis de fikk Telenor, det var liksom basisen. Også har vi tatt samfunnsansvaret vårt alvorlig og forvaltet nettverket vårt og telefonitjenesten, så hver gang vi gjorde endringer, hver gang det ble satt opp prosjekter for å gjøre endringer, så skulle det være full stabilitet og vi kunne ikke ta ned nettet. Så da var det lange analyseperioder, slik at vi var dønn sikre på hva vi skulle gjøre»*

Det at man i Telenor Norge behandler sikkerhetsklarert data gjør ifølge intervjuobjektene at det er farlig å jobbe agilt på enkelte prosjekter. Dette er fordi at ifølge intervjuobjektene kan agile metodikker på feil prosjekt i verste fall føre til feil som kan true Norges sikkerhet.

> *«Telenor har jo systemer som sitter på sikkerhetsklarert data, det er et helt rom man ikke har lov til å gå inn med mindre man har snakket med politiet. Det er veldig, veldig mye høyere sikkerhetskrav. Tjenester som er kritisk for Norges sikkerhet er det farlig å være alt for agile på.»*

Siden da har kundebehovet og den teknologiske utviklingen ført til at Telenor har innsett behovet for å endre de prosessene som driver programvareutvikling. Selv om disse nødvendigvis ikke er like kritisk for samfunnet er det mange av utviklingsprosjektene som fortsatt bærer preg av mye kontroll og tungvinte prosesser. I disse prosjektene tar det gjerne veldig lang tid før selve implementeringen bli gjennomført. Mye av tiden går på analyse i forkant av utviklingen, og prosjektledere skal gjerne ha kontroll over hele utviklingsforløpet før prosjektstart, men både teamet og lederne har et ønske om hurtigere prosesser.

> *«… Generelt er det jo ønskelig at man kommer fortere i gang med implementeringen og ikke analyserer for mye da. Man bruker for lang tid på den analysefasen»*

Blant intervjuobjektene er det stor enighet om at man er nødt til å endre arbeidsmetodikk for å klare å holde seg konkurransedyktige. Ifølge et av intervjuobjektene har man klart å skape en følelse av at det «brenner» under beina på dem i Telenor Norge.

> *«… på Norgesnivå og i teknologidivisjonen syns jeg at vi har klart å skape en følelse av en sånn «burning plattform», nå brenner det under beina på oss. Det kjenner jeg. Så det rasjonale for å endre seg er i hvert fall godt fremlagt og folk skjønner at vi er nødt til å endre måten vi arbeider på. Så vi opplever at det er et slags trykk i organisasjonen på det.»*

Hvilken type arbeidsmetodikk som historisk har blitt brukt i Telenor Norge påvirkes ifølge intervjuobjektene av kompleksitet og usikkerhet til prosjektene. Selv om Telenor Norge prøver å tilpasse arbeidsmetodikken til å kunne levere raskere og til kundens endrede behov ser vi at ledelsen vegrer seg mot å gi fra seg kontrollen til et lavere nivå i organisasjonen. Dette medfører ifølge intervjuobjektene stor grad av dokumentasjon og høye krav for godkjennelser av prosjektene, betingelser som gjør det vanskelig å kunne omstille seg og levere kontinuerlig etter kundens behov.

4.2.2 Agile praksiser

Utfordringen med å implementere agil metodikk i Telenor Norge mener intervjuobjektene kommer som følge av størrelsen på bedriften, og konsekvensen med å gjøre feil. Med mange tusen ansatte må det være kontrollmekanismer for å sørge for at de ikke kan gjøre fatale feil.

> *«Om du har 5 personer som jobber sammen i lite firma, så ville alle kjenne hverandre kjempegodt, hele tiden jobbe tett sammen, vite hvordan man skulle gjøre ting og avtale ting løpende, men når du har en bedrift med mange tusen vil du alltid ha visse kontrollmekanismer på plass for å sikre at de ikke gjør fatale feil.»*

For å kunne tilrettelegge for en mer agil tilnærming prøver prosjektstyringsgruppen i Telenor Norge å dele opp prosjektene i mindre biter. Denne oppdelingen mener flere av intervjuobjektene muliggjør å drive agil utvikling på de deler av prosjektet som ikke berører personvern og sikkerhet.

> *«Da handler det om å være god og bli enda bedre på å stykke opp prosjektet, satsningene, i mindre biter så vi ikke har store klumper som skal leveres som i praksis blir veldig fossefall. Så vi stykker det opp i mange mindre biter så du da klarer å være*

mer agil i leveransen, men vi får også sortert i større grad de delene vi kan være agile på.»

«Men det er også utrolig viktig at personvern og sikkerhet fungerer godt for det er ingen tilgivelse om det kan hackes. Derfor får du sortert ut den type leveranser, som er da kall det vannfall, også får du skilt ut mange andre leveranser hvor det kan jobbes agilt»

Ved å tilrettelegge for agil utvikling der det er mulig klarer teamene å utvikle mer effektivt og innfri målet til ledelsen om raske leveranser. Et av intervjuobjektene, som arbeider med utvikling og forbedring av applikasjoner, mener at frihet til å bestemme arbeidsmetode er en forutsetning for å kunne levere fungerende programvare effektivt.

«… [det er viktig med] makt til utviklerne og la de som vet hvordan ting fungerer bestemme mer enn hva man tradisjonelt har gjort. Og ikke la resten av organisasjonen være en sten i skoa for dem, men heller klare å reise opp og støtte opp om det»

«… vi har på en måte bare kommet frem til hvordan vi kan jobbe mest mulig effektivt, og det er hvis vi kan bestemme mest mulig. Hvis vi som faktisk lager ting kan bestemme mest mulig, legge ut til produksjon når vi vil, så kommer resten av seg sjøl litt sånn etter …»

I enkelte av avdelingene er det ifølge et av intervjuobjektene i stor grad dem selv som velger stor grad hvordan arbeidet skal gjennomføres så lenge arbeidet blir levert til avtalt tid og krav. Det er lite fokus på dokumentasjon og rapportering av arbeidsprosessen og oppfølgning tas muntlig etter behov. Dette fungerer fordi utviklingen i denne avdelingen i stor grad er uavhengig av annen utvikling, og utviklerne i stor grad arbeider for seg selv. Prosjektene i denne avdelingen er også i stor grad sett på som risikofrie, da de ikke berører personvern eller mobilnettet.

«Det er på en måte sjefen som har gitt oss friheten, frihet under ansvar da … så jeg prøver å bruke mest mulig tid på utvikling og det som faktisk behøves for å faktisk kommer frem til løsninga og det produktet jeg er bedt om å lage.»

Kriteriene for å kunne jobbe slik blir satt av prosjektgruppen som vurderer konsekvens opp mot risiko ved å jobbe med et produkt. Hvis konsekvensen blir for høy, får man ikke lov til å benytte agile metodikker på samme måte.

«Et av hovedkriteriene er faktisk litt tekniske, at du ikke kan skape store konsekvenser for Telenor. Om du kan skape store konsekvenser for Telenor i forhold til de tekniske løsningene eller forretningsvirksomhet osv. så kan du ikke kjøre agilt. Skal ikke touche at du kan ta ned mobilnettet»

Dette er rett og slett fordi at Telenor Norge behandler data som er så sensitive at de ikke kan gjøre feil med dem, og agile metodikker blir sett på som «prøv-og-feil» metodikk i større grad enn fossefallsmetodikker blir.

«hvis du implementerer en utdata, skriver bare en linje med kode på et eller annet som implementeres på en database som forteller noe om hvem som eier abonnementet Erna Solberg, så kan kanskje Russland hacke seg inn via den koden uforutsett, altså det var jo ingen som skjønte at den sammenhengen der gjorde det mulig å hacke, som igjen gjør at de får tilgang på livsfarlig informasjon, ja. Det er ikke akseptabelt rett og slett»

Prosjektstyringsgruppen ønsker likevel at de fleste utviklingsavdelingene skal kunne jobbe kunne agilt, men at dette er vanskelig å implementere i avdelinger som jobber med kritiske tjenester, spesielt de som er knyttet opp mot mobilnett og personvern. På spørsmål om i hvilken grad Telenor Norge tilstreber å arbeide agilt svarer et av intervjuobjektene dette:

«Vi gjør det til stor grad der det ligger godt til rette for at konsekvensene ved å gjøre det er små. Der gjør vi det ganske utstrakt vil jeg si. Så det å endre en app, f.eks. Telenor der du kan gjøre endringer på abonnement, sjekke data.. altså de ulike kanalene på Telenors websider jobber vi smidig. Det er ikke så farlig om siden går ned, den får vi opp igjen. Men der vi ikke jobber smidig er hvor vi risikerer at mobilnettet går ned. Det er ganske sign. off fossefalls orientert for å ha kontroll.»

Den tradisjonelle måten å arbeide på vil fortsatt være gjeldende hvor det er behov, slik at det blir et skille mellom de to ulike tilnærmingene for arbeidsmetodikk. Telenor Norge har valgt å kalle prosjekt som skal bruke agil arbeidsmetodikk for røde prosjekt og prosjekt som er mer utsatt for risiko for blå prosjekt. For å imøtekomme denne to-delingen har de også valgt å ha dedikere røde og blå prosjektledere, altså egne prosjektledere for ulike typer prosjekt.

Rød arbeidsmetodikk blir i dag brukt i noen avdelinger, og intervjuobjektene opplever at det er mye fokus på minimumsløsninger, tett kundeinvolvering og eierskap som sørger for hurtig leveranse og feilretting. Et av intervjuobjektene sier at denne tilnærmingen gjør at de slipper å bruke unødvendig mye tid på analysefasen.

«fordelene er at vi kan tidvis ha enormt bra leveransekapasitet og time-to-market, og særlig feilfiksing, fordi vi overvåker jo feil selv, vi har hele verdikjeden ifra start til mål. Så hvis vi ser en feil i produksjon så fikser vi det uten at noen nødvendigvis melder det inn.»

Ifølge flere av intervjuobjektene er Telenor Norge nødt til å skifte fokus. Der fokuset tidligere har vært på at produkter skal være «perfekte», er nå hurtighet og fleksibilitet viktig for å kunne få tilpasset løsningen ut i markedet.

«…Telenor er jo et stort maskineri og vi har jo vært ikke flink til å få ting raskt ut i markedet og når det kommer ut så funker ting bra. Men at man heller fokuserer på, ikke akkurat kvalitet, men at man har mer fokus på å være raskt ute med en løsning enn at løsningen er perfekt, så får man heller gjøre justeringer i etterkant»

For å tilrettelegge for en agil tilnærming har Telenor Norge valgt å dele opp prosjektene i deler hvor det ikke er like stort behov for kontroll og konsekvensene er små. Ifølge intervjuobjektene som jobber rødt har agil metodikk fokus på minimumsløsninger, tett kundeinvolvering. Dette gjør at utviklerne får eierskap til tjenestene, kan reagere raskt på endringer og levert deler av tjenesten hurtig. Samtidig som denne tilnærmingen fører til at Telenor Norge evner å jobbe mer agilt i tjenester med lav risiko, begrenses mulighetene for å jobbe agilt i de resterende tjenestene.

4.2.3 Squads

For å legge mer til rette for agile metodikker i hele organisasjonen, har Telenor Norge implementert et organiseringsrammeverk med team som de kaller «squads», som igjen er organisert i en samling kalt «tribe». Squad er et selvorganisert team som i stor bestemmer selv hvordan de arbeider. Teamene, eller squadene, er tverrfaglig og består følgelig av alt fra domenearkitekter til UX-designere. Dette gjør at squadene kan ferdigstille leveranser uten å være avhengig av andre. Ifølge intervjuobjektene er det bestemt at hver squad skal ha en backlog, og at denne skal dokumenteres i et datasystem slik at man skal få mest mulig transparens i hele organisasjonen. Utenom dette får squadene selv styre hvordan de skal jobbe på team-nivå. Agile, eller røde, prosjektledere og produkteiere styrer squaden og har i samarbeid med kunden kontroll på hva som skal inn i backlog og dermed hvilke arbeidsoppgaver squaden skal arbeide med. Videre vil en tribe bestående av flere squads, og arbeide med bestemte funksjonaliteter som den triben eier og utvikler, f.eks. en nettside..

«Vi driver jo nå og implementerer den nye måten å jobbe på, i 2020 skal Telenor Norge ha gjort om hele IT-organisasjonen til triben med underliggende squads ... Utviklerne skal jo nå, i noen av pilotområdene til Telenor i alle fall, samles i squads hvor man har tverrfaglige team med en arkitekt, en PO, løsningsutviklere, service design utviklere osv.»

I det nye organiseringsrammeverket skal Telenor gå over fra å være prosjektbasert til å bli mer systembasert. Dette vil si at i stedet for at man fokuserer på prosjekter som skal gjennomføres, så skal fokuset ligge på systemer og kanaler. Her skal tribene og squadene få ansvar for hvert sitt system og kanal, og dermed klare å ha mer fokus på å utvikle det og holde det ved like.

«I stedet for at prosjektet er i sentrum for hva man organiserer rundt, så organiserer man seg rundt systemet. Så det betyr at du får en gruppe mennesker som hele tiden har ansvaret for at den kanalen, for eksempel websiden, er up to date»

Ved å innføre tverrfaglige og selvorganiserte squads håper Telenor Norge at dette vil føre til en mer effektiv organisering av ressursene, samt at det legger til rette for å skille systemer man kan jobbe agilt og de som ikke har mulighet. På bakgrunn av intervjuobjektenes tilbakemeldinger ser vi at transparens på tvers av squadene vil redusere avhengigheter og behovet for rapportering.

For å sørge for at disse squadene tilnærmer seg den agile måten å jobbe på har Telenor Norge valgt å sende prosjektlederne som skal ha sin egen tribe på kurs.

«... så gjorde vi det i prosjektlederenheten i mai/juni og utover høsten i et 6 ukers program i regi av INSEAD og en professor der som har lagd et online opplegg med case og innlevering og gruppearbeid for å tilnærme oss denne agile måten å jobbe på»

4.2.4 Oppsummering

Som følge av mange tusen ansatte og risiko i tjenestene de utvikler er det utfordrende å kunne implementere agil metodikk. Likevel tilstrebes dette å skille ut delene av prosjektene hvor utviklerne kan jobbe agilt. Ofte har dette vært i situasjoner hvor utviklere har jobbet uavhengig av andre. I slike tilfeller har det vært lite prosessfokus generelt, i stor grad fordi det ikke har vært behov for det. Nå i nyere tid har man bestemt at et rammeverk som fremmer agil arbeidsmetodikk skal implementeres i hele IT-avdelingen til Telenor Norge. Dette baserer seg på team kalt «squads» og samlinger av team kalt «tribes». Squads er i praksis selvorganiserte

team med lite prosessføringer på teamnivå, men med enkelte føringer om at de skal bruke for eksempel backlog, JIRA og lignende. Denne transparensen i avdelingene bidrar til å redusere avhengigheter og behov for rapportering. For å sørge for at squadene tilnærmer seg agil metodikk har Telenor Norge sendt prosjektlederne på kurs og lagt til rette for oppfølgning. I komplekse prosjekt med høy usikkerhet kommer Telenor Norge til å beholde tradisjonell fossefallsmetodikk fremfor å jobbe agilt ifølge intervjuobjektene. Dette mener Telenor Norge er helt nødvendig for å sikre at systemkritiske tjenester ikke blir utsatt for usikkerheten det er ved å jobbe agilt.

4.3 Metodetilpasning i Telenor Digital

For å finne ut hvordan Telenor Digital i praksis har tilpasset agile metodikker til sitt bruk, ble spørsmål rundt dette stilt i intervju. Både om hva de spesifikt gjør for å forbedre arbeidsmetodikk, hvordan de historisk har gjort det og kommet frem til dagens metodikk, og om de mener noe kunne vært gjort annerledes enn hvordan det gjøres i dag. Det ble også spurt spørsmål om hva som krevdes for å kunne jobbe agilt.

Intervjuobjektene mener at forbedring av egen arbeidsmetode er noe de må gjøre. Dette må gjøres bl.a. for å tilpasse seg *virkeligheten*, og på grunn av at produktet deres og prosjektene deres alltid henger sammen med andre prosjekter rundt om i Telenor.

> *«Vi har vært nødt til å gjøre det, og vi har vært gjennom flere iterasjoner for å drive kontinuerlig forbedringsarbeid av egen arbeidsmetode for å tilpasse oss virkeligheten. Hvis du ser på pure Scrum som det er definert så tar den utgangspunkt i team som opererer i full isolasjon fra andre team uten avhengigheter, som vi aldri gjør i virkeligheten.»*

Et av intervjuobjektene sier også at metodetilpasning er noe av det viktigste med den agile metodikken, og at det er helt nødvendig for å kunne få noe nyttig ut av agile metodikker som de er foreslått i «teorien». Retrospektiver trekkes også frem som viktig for metodetilpasning. Mye av funnene i dette delkapittelet er relatert til nettopp retrospektivene.

> *«Det er jo en del av det jeg oppfatter som er viktigst med den agile prosessen, i hvert fall i praksis at du har en slags evaluering og en tilpasning, hvordan man gjør ting gjennom en retrospektive og da i den emfiserer tilpasninger man kan gjøre, men det er*

også en hel del tilpasningssteg fra den teoretiske modellen til man i det hele tatt gjør noe praktisk.»

4.3.1 Arena for metodetilpasning

Mye av metodetilpasningen i Telenor Digital skjer ifølge intervjuobjektene i retrospektivmøtene. På spørsmål om når og hvordan prosessendring tas opp i Telenor Digital svarer et av intervjuobjektene følgende:

> *«Den generelle måten å håndtere det på er i retrospektivmøter, og det har vi jo hatt hele tiden som en del av prosessen vi har kjørt og det har vært veldig nyttig for å komme opp med ideer.»*

> *«Jeg føler at retrospektivene har vært en veldig god måte å ha bevissthet rundt metodikk, hvordan vi faktisk gjør ting»*

Ifølge intervjuobjektet handler retrospektivene om prosessendringer fra en sprint til den neste. Teamene går gjennom hva som fungerte og ikke i den foregående sprinten, og hva de kan prøve ut eller slutte med i den neste. Endringer følges så opp ved neste retrospektiv, slik at man får en gjennomgående forbedringsprosess.

> *«Sånn konkret nå så er det at vi setter oss ned og stiller noen spørsmål om at vi skal tenke på sprinten som var, sånn at det er veldig sånn fokusert veldig kort tid bakover. At man skal komme opp med ting man skal slutte å gjøre, begynne å gjøre eller fortsette å gjøre – så det er de tre kategoriene vi skal prøve å sortere i … [og så] noterer vi oss konklusjonen fra retrospektiven, og det følges da opp neste gang: har vi virkelig startet med det her, har vi stoppet med det her, har vi fortsatt å gjøre det her.»*

Dette gjør at det er først og fremst mindre endringer som gjøres i retrospektivene, mens større endringer skjer hvis omstendigheter endrer seg, enten i form av for eksempel ferieavvikling eller at nye personer kommer inn.

> *«… men det er begrenset hvor store endringer man kan gjøre i en retrospektiv. Det er på en måte innenfor sprint tankegangen. Scope er på en måte begrenset til den sprinten man har retrospektiv i. Vi har tidligere eksperimentert, i forbindelse med sommerferie og sånn hvor man blir enig om at det ikke fungerer med en sprint, sånn at nå kjører vi Kanban i sommer og sånne ting.»*

«Men det som vi gjør nå at vi har endret mer er delvis fordi det har kommet inn en ny product manager som er en gylden anledning til å gjøre tilpasninger.»

Dette er også noe de andre intervjuobjektene i stor grad sier seg enig i. Intervjuobjektene trekker frem at retrospektivene skal være teamene sin arena for å komme med prosessforbedringer

«Retrospektive er jo teamet sin arena for å diskutere hva som er bra og hva som kan forbedres.»

Forslag til metodikk som teamet har lyst å teste ut er noe som i stor grad kommer fra andre avdelingers eller bedrifters vellykkede erfaring med metodikk.

«Det var noen andre som hadde brukt det her, de som jobber med det andre produktet. Vi hadde hørt rykter om at det var bra.»

Ifølge et av intervjuobjektene er metodetilpasningen og prosessforbedringen i Telenor Digital veldig demokratisk. Det poengteres dog at enkelte deler av arbeidsmetodikken er forbeholdt lederne, noe som står i kontrast til resten av daglige arbeidsmetodikken.

«Jeg føler at det er en veldig demokratisk modell. Igjen så er det som sentrert retrospektivmøte, det er dedikert til prosess, da har alle mulighet til å komme opp med sine synspunkter. Det er klart, noen aspekter ved den totale metodikken er forbeholdt type team lead eller product managers, særlig mye av det med innhenting av arbeidsoppgaver og hvordan det legges ut i litt større planer, det er på en måte en disconnect i forhold til den daglige metodikken alle er en del av.»

Dette er noe vi ser går igjen i svarene fra intervjuobjektene. Intervjuobjektene sier at det er viktig at de får bestemme hvordan de skal jobbe, og at det er rom for å gi beskjed om at denne måten å jobbe på ikke funker hvis det er nødvendig. Det er også fokus på at avgjørelser om arbeidsmetodikk i hovedsak skal tas gjennom konsensus i teamet, og ikke av ledelsen alene. Et av intervjuobjektene sier det på følgende måte:

«Normalt sett vil man ha en konsensus avgjørelse i teamet, hvis det er sånn at teamet ikke klarer å være enig kan sikkert PM'n skjære igjennom til sist, men den sitter ganske langt inne altså.»

Selv i større endringer som tas utenfor retrospektivmøtene føler intervjuobjektene at de blir involvert, dog ikke i like stor grad. Men intervjuobjektene opplever at det er også viktig at det er takhøyde for å ta opp et tema hvis man føler seg diktert.

> *«Jeg føler at det har vært en litt sånn involverende prosess, og da vil du naturlig noen ganger komme opp i en situasjon hvor noen føler seg diktert eller det blir litt konflikter av det. Så det gjelder å ikke være redd for å ta de diskusjonene og konfliktene.»*

Et annet intervjuobjekt påpeker videre at det ikke er alt som er like involverende og demokratisk, noe som tidvis kan skape problemer når man skal planlegge hvilke oppgaver teamet skal gjøre i sprintene.

> *«Prosessen for å legge kvartalsmål eller roadmaps er ikke demokratisk på samme måte som Scrum-biten og det burde den kanskje være, eller hvert fall være samme type åpenhet. Og det har i kanskje i perioder ført til at folk at de egentlig ikke har frihet i sprint planning til å gjøre det man syns er viktig fordi du har commitments eller planer.»*

Et av intervjuobjektene påpeker videre at teamet i større grad enn hva teori tilsier bestemmer hva dem skal arbeide med, fordi *«vi har såpass dyp teknisk innsikt i plattformen vår og vet hva som er riktig»,* noe som altså tidvis kan være vanskelig siden teamet ikke er like involvert i å planlegge kvartalsmål og lignende.

Ifølge intervjuobjektene jobber de med å bygge en kultur der man har lyst til å forbedre arbeidsprosessene sine. Dette kan skje ved å eksperimentere i arbeidsmåten, gjøre små forandringer ofte og være bevisst på hva de gjør.

> *«…. Det er ikke slik at vi snur om på ting hele tiden, men små tweaks gjøres ganske ofte. Og det er på en måte å prøve ting og se om det virker, det er ganske viktig. Hvis vi liker det så fortsetter vi, hvis ikke så slutter vi »*

> *«… det er noe med å jobbe med å bygge en kultur på, være bevisst på hvordan vi jobber og hvordan vi forbedrer oss. Det er viktig elementer med å jobbe agilt.»*

4.3.2 Squads

Det er derfor kontinuerlig fokus på prøve nye agile praksiser, hvor de nylig eksperimenterte med et selvorganisert team. 4 personer ble satt i et eget team kalt en «squad» og fikk beskjed om å jobbe med et avgrenset prosjekt i ca. 1 og en halv måned. Nøyaktig hvordan de skulle

jobbe innad i teamet fikk de beskjed om å organisere selv. Dette er en type arbeidsmetodikk som kommer fra Telenor Norge, og som i stor grad består av kryssfunksjonelle selvorganiserte team. Når man jobber i squads var det ifølge intervjuobjektene et poeng at man skulle ha en godt avgrenset tjeneste eller system å jobbe mot. Dette skulle gjøre at man kunne jobbe uavhengig av andre, og på den måten unngå hindringer.

> *«Det ble presentert for oss at dette var noe vi kunne prøve ut, sånne Squads. Da var det snakk om å trekke ut noen fra frontend og backend teamet for å jobbe med en spesifikk ting for å ha fokus på en tjeneste/lite produkt som var avgrenset slik at man ikke var avhengig av at andre utenfor squaden skulle gjør ting, for å unngå blokkers.»*

Squaden fikk stort sett beskjed om å organisere seg selv. Dette var et prøveprosjekt, så tidsperioden var ganske avgrenset. Men ifølge intervjuobjektene var det vellykket, og noe de kom til å prøve igjen i fremtiden.

> *«Samme dag, som var rett før sprinten skulle begynne så fikk vi beskjeden: ok, fra i morgen jobber dere med Squaden- organiser dere selv. Det var kanskje en og halv måned vi fikk fram til jul.»*

4.3.3 Oppsummering

Metodetilpasning i Telenor Digital er i stor grad sentrert rundt retrospektivmøtet. Dette er et møte som er dedikert til metodetilpasning og prosessforbedring, og som ifølge intervjuobjektene fungerer godt. Live-teamet er et eksempel på en prosessforbedring som er kommet ut fra retrospektiv, og er ifølge intervjuobjektene nyttig for å få tatt unna hasteoppgaver. Det er også verdt å merke seg at det er rom for å *ikke* benytte slike forslag til prosessforbedringer hvis man ikke ønsker det, noe som kommer frem ved at bare ett av de to teamene har tatt i bruk live-team. Dette forklares med at de synes det tar bort for mye av fokus fra sprinten, og trekker også frem at det er viktig at de selv får bestemme hvordan de skal jobbe. Dette sier de andre intervjuobjektene seg enig i, og det er tydelig at er et fokus på å skape en arena hvor man ønsker å forbedre prosess samt å prøve nye metoder. Et konkret eksempel er et eksperiment på å opprette en «squad», som i praksis minner veldig om et selvorganisert team.

4.4 Metodetilpasning Telenor Norge

For å finne ut hvordan Telenor Norge tilpasser agile metodikker ble personer fra prosjektstyringsgruppen, prosjektledere og utviklere intervjuet. Det ble stilt spørsmål om hva de spesifikt gjør for å forbedre arbeidsmetodikk, hvordan de er kommet frem til dagens metodikk, og om de mener noe kunne vært gjort annerledes enn hvordan det gjøres i dag.

4.4.1 Arena for metodetilpasning

På overordnet nivå er det ifølge intervjuobjektene prosjektstyringsgruppen som har ansvaret for å finne ut hvordan de kan forbedre arbeidsmetodikken. For å avdekke hvordan de skal gjøre det ser gruppen på hvilke problem som er reelle for å så trekke til seg personell med kompetanse på fagfeltet for å gjøre vurderinger. Dette gjøres fortløpende, men en gang i året blir det arrangert et ledelsesmøte hvor prosjektledere blir samlet for å gå igjennom ulike tema og vurderinger.

> *«… vi har ansvar for å forbedre oss og finne en bedre måte å levere prosjekter på i fremtiden. Så vi har et forbedringsansvar. Har ansvaret for at de konkrete prosjektene blir kjørt riktig, slik at vi finner de rette leveransemodellene for prosjektene»*

Videre sier et av intervjuobjektene at Telenor Norge har en hovedmodell for hvordan man skal levere prosjekter, basert på den er det prosjektledernes ansvar for å velge riktig metode. Disse metodene blir gruppert etter hvorvidt prosjektet skal kjøres etter agile eller vannfallsbaserte metoder.

> *«I Telenor [Norge] har vi en hovedmodell for hvordan man skal levere prosjekter og innenfor den så skal man velge riktig metode for å levere deler av prosjektet/leveranse. Det kan være smidig leveransemetode eller det kan være fossefallsmetode. Så det er prosjektleders ansvar å velge riktig metode.»*

Men for prosjektlederne er det ifølge intervjuobjektene utfordrende å skulle tilpasse metoden uten å vite hvordan dette kan gjøres. Uklar rollefordeling og ansvar for hvilke verktøy og metoder som skal benyttes gjør at det blir vanskelig å tilpasse seg etter kravene.

> *«Våre prosjektledere slites nok litt mellom den faste måten å jobbe på, også er virkeligheten helt annerledes nå, også prøver vi da å tilpasse måten å jobbe på til den nye virkeligheten. Også har vi ikke kanskje så veldig gode metoder og verktøy for det, enda».*

For å være mer tydelig hvordan de kan være agil i prosjektene har prosjektstyringsgruppen valgt å innføre en ordning med dedikerte agile prosjektledere. Tidligere har det ikke vært så tydelig at Telenor Norge vil jobbe agilt, og ifølge intervjuobjektene har dette ført til at prosjektlederne har møtt et veldig «fossefall-Telenor» som ikke har lagt til rette for agile metodikker.

> *«... akkurat i disse dager lager vi en tittel: agile project manager. Der noen skal være agile managers og være god på det. Vi starter akkurat nå på å formalisere mer roller og krav til den rollen ift hva du må være god på, og neste steg metode hvordan du skal levere agilt. Foreløpig har det vært fleksibelt. For noen har det vært helt toppers, men for andre har det kanskje vært litt for utydelig og følt at de har møtt et veldig fossefall Telenor som ikke er tilrettelagt for å kunne jobbe agilt, og vært litt frustrert. Så nå skal vi være litt mer tydeligere i neste steg».*

Prosjektledere som skal jobbe agilt blir opplært gjennom kurs og foredrag for å sikre en mer formell tilnærming til agil arbeidsmetodikk.

> *«… for mitt vedkommende så gjorde vi det i prosjektlederenheten i mai/juni og utover høsten i et 6 ukers program i regi av INSEAD og en professor derfra …»*

På team-nivå, altså internt i squadene, sier et av intervjuobjektene at de tar i bruk for eksempel retrospektiv og et testrom som de kaller service design lab for å kunne vurdere nye løsninger internt og opp mot kundene. Det legges også til rette for at internt i squadene er det squadene selv som bestemmer hvordan de vil tilpasse metodikkene sine. Retrospektiv er en mye brukt metode, men ifølge intervjuobjektene er det ikke noe krav om å benytte det. Denne tilnærmingen mot agil arbeidsmetodikk vil ifølge intervjuobjektet spare avdelingene for unødvendig ressursbruk og sikre kompetanseoverføring.

> *«Og det er veldig viktig at du har tillit internt i gruppa og sparrer om ting gjennom retrospektiv hvor de utfordrer hverandre på en konstruktiv måte sånn at de hele tiden drar det videre.»*

> *«… vi har bygd vårt eget testrom, det er derfor den sonen her heter service design lab hvor hele prosjektet kan sitte å følge med på hva er det kunden faktisk sier, hvordan er det de faktisk reagerer.»*

Ifølge intervjuobjektene skal det nye organiseringsrammeverket legge mer til rette for bruk av både agile metodikker, men også fossefall der det trengs. Dette gjør at man få et slags to-delt

fagmiljø, der det ene skal bli gode på å jobbe agilt, mens det andre skal bli gode på å jobbe tradisjonell fossefall. Det at man blir to tydelige fagmiljøer vil ifølge et av intervjuobjektene gjøre det enklere å lære av hverandre, noe som anses som viktig i Telenor Norge.

> *«Vi ser at vi må lære av hverandre og kanskje det er bedre å ha en gruppe da, som jobber på den måten og som kan lære av hverandre og være litt tettere knyttet i stedet for at man må jobbe litt på tvers og søke opp de folkene eller hva jeg skal si, så kan det være greit å ha to sånn fagmiljøer på en måte. Hele organisasjonen i teknologidivisjonen blir jo satt opp mer nå for å jobbe på den måte"*

I avdelinger der de har jobbet uten mye prosessføringer og bestemte metodikker, har det etter hvert utviklet seg enkelte normer. Dette er ifølge intervjuobjektene fordi man kjenner hverandre så godt etter hvert.

> *«Vi har noen normer som, det er sånn stille enighet om da, men vi har ikke noe nedskrevet. Det er bare at vi kjenner hverandre godt. Og vi lager APIer på en gitt måte, og vi snakker jo en del sammen selvfølgelig sånn at, de normene er der, men de er jo ikke notert»*

Intervjuobjektene trekker også frem at for å virkelig kunne jobbe agilt, må tankegangen i organisasjonen endres. Et av intervjuobjektene mener at man i stedet for å feire at man lanserer et produkt, så burde man kunne ferie når man lærer noe. Være det seg at man lærer at kunden vil ha produktet, eller at kunden ikke vil ha det i det hele tatt.

> *«tradisjonelt så feirer jo både vi og veldig mange andre selskap når man lanserer ting. Så da har du lanseringsfest, og lanseringskake og lanseringsballonger. Det betyr at når du som prosjektleder setter i gang med et prosjekt så er du veldig opptatt av å lansere. Fordi da får du ballongene dine og kaka di. Men du får det ikke ellers. Og da er det ikke mulig å jobbe verken agilt eller design thinking eller noen ting. Så de jeg heller vil at vi erstatter det med er at vi feirer læring. Og da handler det om at vi kan si at, «wow, nå har vi lært at kunden vi ha dette her, så kan begynner vi med dette, og underveis lærte vi at vi ikke skal bygge det sånn men vi skal bygge det sånn».*

Dette vil ifølge intervjuobjektet gjør at man kan jobbe *genuint* agil, ved at det legger mer til rette for en «prøv-og-feile» kultur. Ifølge intervjuobjektet skal man ikke bare kunne feire at man lærer seg hvordan ting skal gjøres, det skal legges like mye vekt på å lære hvordan ting *ikke* skal gjøres, som for eksempel at et produkt ikke har det potensialet man tidligere trodde.

Ifølge intervjuobjektet er kanskje det en enda viktigere del for å legge til rette for genuin agil arbeidsmetodikk.

> *«… Og så kan du ha kake og ballonger for at du slutter å gjøre dette her. Den siste der tror jeg er ganske viktig for selskap som har lyst til å jobbe genuint agilt.»*

4.4.2 Oppsummering

Historisk har metodetilpasning i Telenor Norge skjedd gjennom kontinuerlig forbedringsarbeid. Nå som de har bestemt seg for å tilrettelegge for agile metodikker i større grad, har prosessene rundt metodetilpasning blitt mer formalisert. Det hele styres i stor grad av en prosjektgruppe som har ansvar for implementering av organiseringsrammeverket. Dette kan ses i en kontrast mot hvordan det gjøres på team-nivå, der det er mindre formalisert. Her benyttes gjerne retrospektiv på samme måte som i Telenor Digital, eller andre prosesser. Dette er noe teamene, eller squadene, skal styre selv ifølge intervjuobjektene. Det er også tydelig at Telenor Norge mener at en arena for læring er viktig for å få til å jobbe «genuint agilt».

5 Diskusjon

I dette kapittelet skal vi drøfte funnene fra empirikapittelet opp mot relevante teorier fra teorikapittelet. Kapittelet er bygd opp etter de to forskningsspørsmålene, hvor diskusjon og tolkning vil danne grunnlaget for konklusjonen i kapittel 6. Besvarelsen av det første forskningsspørsmål vil avdekke hva som påvirker hvordan Telenor Norge og Telenor Digital benytter agile metodikker, og det andre forskningsspørsmål vil belyse hva som påvirker hvordan de tilpasser de. Sammen vil disse spørsmålene besvare problemstillingen i sin helhet.

5.1 Bruk av agile metodikker

I det første forskningsspørsmålet ser vi nærmere på hva som påvirker hvordan Telenor Norge og Telenor Digital benytter agile metodikker. For å avdekke dette deler vi diskusjonen inn i usikkerhetsstyring, agile tilnærminger og kontroll eller agilitet? Forskningsspørsmålet som diskuteres i dette delkapittelet lyder som følger:

> *«Hva påvirker hvordan Telenor Norge og Telenor Digital benytter agile metodikker for programvareutvikling?»*

5.1.1 Usikkerhetsstyring

Telenor har lenge vært en betydningsfull IT-leverandør av kritiske tjenester for Norges befolkning, og i løpet av årene bygd opp tunge og sikre systemer som sørger for dette, men de siste årene har IT bransjen opplevd en stor omveltning. Teknologiske nyskapninger og kunders raske adopsjon av digitale tjenester har resultert i en økt etterspørsel for oppdaterte tjenester. Ifølge funnene våre er det dette som har vært drivkraften i Telenors behov for å endre seg, og nå har det skapt en følelse av at det «brenner» under beina på dem. Samtidig blir systemenes pålitelighet kompromittert av globale sofistikerte trusler som kan skape betydelig skade for Telenors kunder. Hvordan en bedrift evner å tilnærme seg usikkerhet blir derfor vurdert for å være en av de viktigste faktorene til om tjenestene lykkes eller ikke (Gregory et al., 2015, Aakre og Scharning, 2013). Både Telenor Norge og Telenor Digital må forholde seg til usikkerhet, men i varierende grad. Dette gjør at de også tilnærmer seg usikkerhet ulikt, noe vi nå skal se nærmere på.

Aakre og Scharning (2013) skiller mellom konkrete og åpne prosjekt, noe som i stor grad handler om omfanget av oppgavene er kjent eller ikke. Dette mener vi kan relateres til hvordan Telenor Norge velger å tilnærme seg usikkerhet. I empirien kommer det frem at ledelsen i

Telenor Norge etterstreber å skape forutsigbarhet hvor det er nødvendig for å minimere usikkerhet. Dette har tradisjonelt blitt gjort ved at prosjektstyringsgruppen avklarer kravspesifikasjoner og betingelser før prosjektleder velger riktig metodikk i henhold til førende krav og betingelser. Denne fremgangsmetoden innebærer grundig analyser og dokumentasjon, noe som minner om fasene i Difis prosjektveiviser (Difi, 2018), og som Schartum og Sørensen (2014) beskriver som fossefallsmetoden hvor krav og betingelser må oppfylles før produktet kan gå videre til neste fase. Aakre og Scharning (2013) sier at dette gjøres i prosjekt hvor ledelsen har et stort kontrollbehov, og hvor tilstrekkelig informasjon og forutsigbarhet sikrer at løsningene blir utviklet i henhold til ledelsens krav om sikkerhet. Dette mener vi kan ses i sammenheng med behovet for usikkerhetsstyring som er en gjennomgående faktor i flere av Telenor Norge sine systemkritiske tjenester som omhandler personvern og sikkerhet. Ved å legge inn krav om dokumentasjon og betingelser for utviklingsprosessen prøver ledelsen å sikre forutsigbarhet knyttet til kvalitet, tidsplan og kostnad. Dette er noe som er svært tidkrevende, et moment som er gjennomgående i tradisjonell programvareutvikling (Maruping et al., 2009, Fontana et al., 2015, Dingsøyr et al., 2012). Denne måten å styre prosjekter har tidligere vært den rådende praksisen i Telenor Norge, og har fungert bra da fremtiden var mer forutsigbar og det ikke oppstod endringer i behov på samme måte som i dag. I empirien kommer det frem at Telenor Norges «nye virkeligheten» gjør at den tradisjonelle måten å gjennomføre prosjektene på fører til forsinkelser, økte kostnader og et produkt som ikke samsvarer med kundens behov. De samme utfordringene opplevde Highsmith og Cockburn (2001) i tradisjonell programvareutvikling, utfordringer øker i takt med større og mer komplisert teknologi.

Telenor Digital har i motsetning en mer uformell tilnærming til usikkerhet hvor prosjektkravene ikke alltid defineres i forkant av prosjektet, men fortløpende gjennom bruk av agile metodikker. Det legges i større grad til rette for at agile team kan håndtere usikkerhet og respondere på endringer som oppstår underveis, i stedet for å prøve å planlegge for det. Det å håndtere usikkerhet i stedet for bare å minimere den er ifølge Dönmez og Grote (2018) et sentralt moment i usikkerhetsstyring, og noe som Telenor Digital oppnår gjennom en rekke praksiser. Uten den samme rigide prosessen til dokumentasjon og krav blir det enklere for teamene å selv avgjøre hvilken tilnærmingen til prosjektene som er mest hensiktsmessig, og ved å bringe ned beslutningsmyndigheten til et operasjonelt nivå, øker hastigheten og presisjonen i problemløsningen (Moe et al., 2008). Utfordringen for Telenor Digital derimot er å dele opp komplekse oppgaver som har avhengigheter mellom hverandre. Som intervjuobjektene påpekte

vil store og komplekse oppgaver som regel ha avhengigheter med andre oppgaver som det må tas hensyn til. Dette gjør at teamene blir mer avhengig av planlegging og koordinering i prosjektene sine, noe som Dönmez og Grote (2018) hevder er viktig for å unngå tvetydighet og usikkerhet. På bakgrunn av dette mener vi at komplekse oppgaver som skal deles opp krever større grad av koordinering enn de agile metodikkene legger til rette for, og at det i større bedrifter med mer komplekse oppgaver enn Telenor Digital er større behov for planlegging og koordinering, noe som gjør det vanskelig å benytte agile metoder.

5.1.2 Agile tilnærminger

Ifølge intervjuobjektene øker kompleksiteten i oppgaver «eksponentielt» med størrelsen på prosjektene, en utfordring som Petersen et al. (2009) mener gjør det vanskelig å definere alle prosjektkrav i forkant. En av de største fordelene med agile metodikker er ifølge Schartum og Sørensen (2014) at man får brutt ned kompleksiteten i mindre deler og at feil og mangler kan avdekkes tidligere i prosessen. I empirien kommer det frem at Telenor Digital gjør dette ved å jobbe i 2-ukers sprinter hvor oppgavene har ett eller få tema som gjør det enklere for teamet å holde fokus. Ifølge intervjuobjektene var 2 eller 3-ukers sprinter optimalt, da det var for vanskelig og komplisert å planlegge for 4 uker, og med bare 1 uke gikk for mye tid bort til møter og lignende. Dette stemmer i stor grad overens med Scrum som beskrevet av Sutherland og Schwaber (2017), som poengterer at lengre sprinter øker kompleksitet og usikkerhet. Ved å implementere delleveranser som en del av arbeidsmetoden kan man håndtere hver del som et mini-prosjekt med hurtig tilpasninger, og på den måte minimere kompleksiteten og lettere håndtere usikkerhet underveis. Videre benytter TD daglige statusmøter gjennom standups hvor teamet og lederne går igjennom oppgavene, en praksis som teamet til tider opplevde som unødvendig, men nyttig når man trengte innspill til problemer. Dette er noe vi mener kan være med på å bryte ned kompleksiteten som kan oppstå når oppgavene får avhengigheter mellom seg, da det gjør det enklere for teamet å koordinere seg imellom. Standups er noe vi tror vil bli viktigere etter hvert som oppgavene blir større og mer komplekse, da dette per i dag er den eneste arenaen som legger til rette for koordinering innad i teamet.

Arbeidsmengden til hver oppgave estimeres for å få vite om de er små nok til å ta med inn i sprinten. Denne praksisen mener intervjuobjektene gjør at de hyppigere kan levere produkt ut til kunde, samt at det er enklere å levere hva kunden faktisk vil ha. Dette er sammenfallende med hva Koskela og Abrahamsson (2004) sier om bruk av agile metodikker, som påpeker at hyppige mindre leveranser er en god måte å skape fleksibilitet, noe intervjuobjektene mente var

noe av de viktigste med agile metodikker. For å planlegge oppgavene som skal inn i neste sprint benyttes sprint planning for å sørge for riktig prioritering og fremgangsmåte, og her har også teamet mulighet til å påvirke hvilke oppgaver som skal gjøres i påfølgende sprint. Dette er noe vi mener bidrar til å skape fleksibilitet, samt gjør det enklere for teamet å håndtere usikkerhet omkring oppgavene da man får vurdert oppgavene fra flere synspunkt. Ifølge Moe et al. (2008) er det å bringe beslutningsmyndighet ned til operasjonelt nivå en god måte håndtere usikkerhet, samt at det bygger tillit og engasjement. I selve sprinten blir uforutsette oppgaver som dukker opp håndtert av et dedikert «live-team». Dette mener vi er et tiltak som fremmer fleksibilitet, og gjør det mulig for Telenor Digital å holde fokus gjennom sprinten samtidig som de får håndtert endringer fortløpende. Det å benytte agile praksiser som et live-team mener vi er en god måte å håndtere usikkerhet, da tillit til at live-teamet klarer å løse uforutsette oppgaver gjør at man i større grad slipper å ta høyde for det i forkant.

I likhet med Telenor Digital benytter også Telenor Norge agile metodikker, men bare i utvalgte prosjekt og avdelinger. Dette er ifølge intervjuobjektene fordi at den høye graden av usikkerhet og risiko i mange av prosjektene til Telenor Norge krever at de jobber mer kontrollert, noe som ofte resulterer i tradisjonell fossefallsmetodikk. I noen avdelinger legges det til rette for at teamene selv kan bestemme hvilken arbeidsmetodikk som skal benyttes, mens i andre skal det benyttes arbeidsmetodikker med større grad av kontroll i form av predefinert og sekvensielle faser. De samme fasene som ifølge Singh og Soni (2011) gjør at fossefallsmodellen er krevende og lite egnet for hurtig utvikling. Dette mener de må skje gjennom en rekke lette praksiser som legger til rette for hurtig utvikling, også kjent som agil metodikk. Ifølge empirien forsøker Telenor Norge å tilrettelegge for dette ved å dele opp prosjektene hvor konsekvensene av å jobbe agilt er små. I de prosjektene hvor agil metodikk er blitt tatt i bruk ser vi i likhet med intervjuobjektene at fokus på minimumsløsninger, tett kundeinvolvering og et ende-til-ende ansvar sørger for hurtig leveranser og hurtig feilretting.

For å kunne legge til rette for mer bruk av agile metodikker har Telenor Norge implementert et organiseringsrammeverk i hele IT-organisasjonen som baserer seg på selvorganiserte, kryssfunksjonelle team kalt «squads». Ifølge Strand og Karlsen (2014) og Moe et al. (2012) vil slike team kunne løse oppgaver raskt og fleksibelt, uten de samme krav til dokumentasjon og analyse, men det vil samtidig kreve høy teameffektivitet, koordinering av aktiviteter og god arbeidsfordeling. Dette mener vi at de prøver å oppnå ved å gå over fra det de kaller prosjektorientert til systemorientert. Dette vil si at i stedet for at squadene skal jobbe med

bestemte prosjekt, skal de jobbe med bestemte system, som for eksempel nettsiden. Dette skal legge til rette for raskere feilretting, større ansvarsfølelse og mindre avhengigheter opp mot andre system. I denne casen ser vi at det i stor grad er prosjektlederne som skal sørge for at squadene får implementert en agil metodikk slik at Telenor Norge unngår utfordringene med tradisjonell fossefallsmetodikk, men at det er opp til squadene selv å bestemme hvilke praksiser de vil benytte. Dette mener samsvarer med det Collins (2017) og Augustine (2005) mener hensikten med agil prosjektledelse er, hvor lederne skal tilrettelegge for at teamet selv utarbeider metodikken i stedet for å pålegge en bestemt metodikk.

5.1.3 Kontroll eller agilitet?

Telenor Norge og Telenor Digital har tilnærmet seg bruk av agile metodikker på forskjellig måter. I Telenor Digital brukes det tilsynelatende utelukkende agile metodikker, og fokuset er å håndtere usikkerhet gjennom å skape fleksibilitet og legge til rette for å effektiv problemløsning. Dette gjøres ved blant annet ved å benytte live-team, la teamet bestemme mest mulig selv, samt å fokusere på å dele opp oppgaver slik at de blir håndterlige og minst mulig kompliserte. Utfordringen til Telenor Digital er når oppgavene blir større og får avhengigheter opp mot andre oppgaver, som gjør dem mer komplisert og vanskeligere å løse. Dette gjør at det kreves større grad av koordinering innad i teamet, noe som kan gå på bekostning av fleksibilitet.

I Telenor Norge har tidligere fokuset vært i stor grad å kontrollere usikkerhet mest mulig. På grunn av type tjenester Telenor Norge må ha kontroll over, har det ofte vært lite rom for bruk av agile metodikker. Etter hvert som IT-bransjen har endret seg, har Telenor Norge likevel i likhet med Fontana et al. (2015) og Moe et al. (2012) anerkjent behovet for å håndtere usikkerhet gjennom fleksibilitet og tilpasning i motsetning til å bare prøve å kontrollere den. For å kunne tilnærme seg større grad av fleksibilitet og tilpasning uten at systemkritiske tjenester blir utsatt for ytterligere usikkerhet, har de derfor tilrettelagt for en kombinasjon av agil og tradisjonell metodikk, lik det Haffke et al. (2017) omtaler som bimodal IT. I de tjenestene som vurderes for å være systemkritisk benyttes det de omtaler som blå metodikk, kjent som tradisjonell fossefallsmetodikk, for å kunne kontrollere prosessen. I de delene som ikke er knyttet opp mot de systemkritiske tjeneste vil de benytte det de kaller rød arbeidsmetodikk, med selvorganiserte squads som kan benytte de agile metodikkene de mener fungerer best. Denne tilnærmingen sørger for at Telenor Norge tilrettelegger for bruk av agile metodikker, samtidig som de kan benytte en større grad av tradisjonelle metodikker i prosjekter med høy usikkerhet og høye konsekvenser ved feil, og dermed forsøke å sikre forutsigbarhet

og kontroll. Denne kombinasjonen mener vi gir Telenor Norge mulighet til å velge arbeidsmetodikk etter behov som gjør at de får utnyttet fordelene til begge metodikkene samt unngår dens negative aspekter.

Den røde arbeidsmetodikken i Telenor Norge kan på mange måter sammenlignes med tilnærmingen Telenor Digital benytter, hvor det å være fleksibel og respondere på endringer står sentralt. Ved å gå over til en to-delt tilnærming, går Telenor Norge over til å forsøke å håndtere usikkerhet i stedet for bare å minimere den, noe som har vært fokuset tidligere. Den to-delte praksisen mener vi er et stort steg på veien til å bli mer agile, samtidig som det legger til rette for å ikke miste kontroll over kritiske tjenester i selskapet. Beslutningen med å gå til å være mer systembasert i stedet for prosjektbasert er også noe vi mener er en god løsning, da det løser mange av problemene vi ser Telenor Digital har knyttet opp mot avhengigheter og kompleksitet. Fremover mener vi det vil måtte tilrettelegges en større grad av koordinering og planlegging i Telenor Digital enn det er i dag, etter hvert som prosjektene blir større og mer kompliserte. Alternativt kan Telenor Digital løse dette ved å benytte samme tilnærming som Telenor Norge, da det vil gjøre at teamene kan løse oppgaver mer uavhengig av andre, men dette er noe vi tror vil bli vanskeligere etter hvert som størrelsen på prosjektene økes.

5.2 Metodetilpasning

I dette delkapittelet skal vi se nærmere på forskningsspørsmål 2: *Hva påvirker hvordan Telenor Norge og Telenor Digital tilpasser agile metodikker for programvareutvikling?*
Delkapittelet er strukturert opp slik at først kommer vi til å diskutere tilnærmingen til henholdsvis Telenor Digital og Telenor Norge, før vi ser på forskjellene dem imellom og hvorfor det er slik.

Conboy (2009) poengterer at metoder og praksiser som fremstår agile ikke nødvendigvis er det i et gitt prosjekt. Det antydes videre at metodetilpasning er nødvendig for å opprettholde agilitet. Dette er sammenfallende med Turk et al. (2002) som argumenterer for at graden av agilitet i en utviklingsprosess kan defineres på bakgrunn av prosjektteamets evne til å dynamisk tilpasse seg prosessen basert på endringer i miljøet. Derfor kan det tenkes at høy grad av agilitet er i stor grad avhengig av tilpasningsdyktige metoder og erfarne utviklere som vet hvordan de skal tilpasse praksisene effektivt. Tilsvarende argumenterer Henderson-Sellers (2005) for at agile praksiser innebærer muligheten til å tilpasse og finjustere utviklingsmetoder etter behov, samt håndtere kravendringer. De hevder at agilitet i praksis innebærer mer enn å følge et sett av forhåndsdefinerte agile metoder, og krever aktivt tilsyn og tilpasning av disse for å kunne være

agil. Det faktum at metodene bare dekker visse aspekter betyr at bare en metode ikke er i stand til å dekke alle behovene til et prosjekt. Dette er drivkreftene til metodetilpasning som vil diskuteres nærmere i de påfølgende avsnittene.

5.2.1 Retrospektiv

Tilnærmingen til metodetilpasningen i Telenor Digital var i stor grad basert på kontinuerlig prosessforbedring. De hadde utgangspunkt i et lett-vekts Scrum-rammeverk, med 2-ukers sprinter, møter som refinement, planning, review og retrospektiv, og hjelpemidler som backlog og user-stories. Alt dette er metodefragmenter hentet fra Scrum-rammeverket til Schwaber og Sutherland (2017), som er nærmere beskrevet i kapittel 2.3.3. Utover dette ble metoden forbedret etter hver sprint gjennom retrospektiver, hvor mulige endringer ble vurdert i retrospektivmøter og prøvd ut i neste sprint. En tilnærming som knyttes opp mot det å kombinere ulike metodefragment til en fullverdig metode, lik det Campanelli og Parreiras (2015) og Conboy og Fitzgerald (2010) kaller method engineering. Denne fremgangsmetoden påpeker Nelson (2010) er hensiktsmessig for å kunne evaluere prosjektets prestasjon, utvinne erfaringer og videre anbefalinger til neste sprint. Flere av intervjuobjektene påpekte at retrospektivmøtene tilrettela for å oppnå disse fordelene, men at potensialet ikke ble utnyttet fordi perspektivet i retrospektivmøtet var begrenset til den forrige sprinten slik at endringene ofte ble små prosessforbedringer fremfor større. Dette kan ses i sammenheng med det Lyytinen og Robey (1999) mener er mangel på læring, en av nøkkelfaktorene til at IT-prosjekter mislykkes. Fokuset til retrospektivene var i stor grad rettet mot en kort tidshorisont, noe som ikke legger til rette for læring. Denne mangelen på et bredere perspektiv gjør at det kan være vanskelig å oppnå det (Morgan, 2004) kaller dobbelkretslæring. Et viktig moment i metodetilpasning da de bidrar til å identifisere problemer, høste læring og forbedre prestasjoner basert på erfaringer (Myllyaho et al., 2004). I Telenor Digital har mye av fokuset vært på å effektivisere møtene mest mulig, da de ofte blir sett på som tidstyver i hverdagen. Fokuset deres på hurtig læring fremfor mest mulig læring mener vi derfor forhindrer deres mulighet til å kunne utnytte fordelene som kommer av dobbelkretslæring.

En konkret tilpasning som Telenor Digital gjorde på bakgrunn av retrospektiv var et live-team for å ta seg av uforventede oppgaver som hastet, men som ikke var tatt med inn i sprinten. Dette er i utgangspunktet noe som ikke er en del av Scrum-rammeverket, men altså noe intervjuobjektene sier de har hentet fra Kanban. Ved å opprette et live-team fikk Telenor Digital håndtert uforutsigbare hendelser samtidig som sprinten fikk fortsette uten forstyrrelser, en

aktivitet som anbefales av Dönmez og Grote (2018) for å kunne håndtere uforutsette oppgaver. Ifølge Schwaber og Sutherland (2017) er det å benytte bare deler av Scrum, eller Scrum sammen med metodefragment fra andre metoder, ikke en anbefalt fremgangsmåte. Likevel var det enighet blant intervjuobjektene om at det viktigste med å jobbe agilt var å være fleksible og evne å respondere på endringer, et moment Collins (2017) og Augustine (2005) trekker frem som noe av det viktigste agil prosjektledelse skal bidra til. Et live-team som kan ta unna uforutsette oppgaver mener vi derfor er med på å legge til rette for fleksibilitet og øke evnen til å håndtere usikkerhet, og noe som har gjort at Telenor Digital har blitt mer agile.

5.2.2 Kontinuerlig forbedring

Telenor Digital benyttet seg ikke av en dedikert metodeingeniør for å utføre metodetilpasningen, noe Fitzgerald et al. (2000) hevder er nødvendig for at fremgangsmåten skal være vellykket. Det kan likevel argumenteres for at hele teamet fungerte som en slags metodeingeniør, da teamet i retrospektivmøtene utførte samme funksjon som Fitzgerald et al. (2000) beskriver som nødvendig. Om et nytt metodefragment skulle tas i bruk, ble dette ifølge intervjuobjektene stort sett bestemt gjennom konsensus i teamet eller gjennom avstemning på retrospektivmøtene. Det kan derfor tenkes at Telenor Digital har en mer demokratisk tilnærming til metodetilpasning enn hva som foreslås av Conboy og Fitzgerald (2010) og Campanelli og Parreiras (2015); i stedet for at en dedikert metodeingeniør bestemte hvordan man skulle endre arbeidsmetodikken, ble dette bestemt gjennom demokratisk avstemning innad i teamet. Dette er noe flere av intervjuobjektene anser som viktig, da de hevder å vite hva som fungerer og ikke med tanke på arbeidsmetodikk.

Telenor Digital benyttet heller ikke noen form for metodebase. I stedet var forslagene til metodefragment de kunne benytte seg av i stor grad basert på andre avdelinger eller bedrifters erfaring med vellykket innføring av agile metodikker. Fitzgerald et al. (2000) sier at om metodetilpasning skal fungere er det nødvendig med en metodebase som representerer de ulike metodefragmentene. Dette er i kontrast med den uformelle tilnærmingen Telenor Digital benytter. Med en dedikert metodebase kunne man fått større fleksibilitet og spillerom for å prøve nye metodikker og løse nye problemer. Erfaringene til intervjuobjektene med agile metoder var relativt begrenset, selv om de fleste hadde kjennskap til Scrum og Kanban var det få som hadde bred erfaring med agile metoder. Det at de i dag stort sett baserte seg på tidligere erfaringer er et funn som samsvarer med påstanden til Fitzgerald et al. (2006), som hevder at metodetilpasning ofte skjer basert på tidligere erfaringer og uten dedikert metodebase og

metodeingeniør. Men i motsetning til Conboy (2009) hevder de at method engineering godt kan være vellykket uten disse faktorene. Dette er sammenfallende med våre funn, da metodetilpasningen til Telenor Digital ser ut til å være vellykket på flere måter, uten at de i direkte forstand har benyttet seg av verken metodeingeniør eller metodebase. Likevel kan det tenkes at en dedikert metodebase vil kunne forenkle arbeidet med å finne og systematisere ulike agile metodefragment. Dette kunne gitt de ansatte en bredere innsikt i hvilke metoder som er hensiktsmessig å benytte til gitt kontekst, spesielt i større organisasjoner hvor det vil være tidkrevende å overføre og oppdatere erfaring mellom alle ansatte.

Dette kan også knyttes opp mot mangelen på kjennskap til agile metoder. Som det kommer frem i empirien hadde ingen av intervjuobjektene i særlig grad kjennskap til agile metodikker. På bakgrunn av den mengden erfaringen som kreves for å lykkes med contigency factors er det derfor nærliggende å tro at det ikke er mulig for Telenor Digital å benytte seg av denne type tilnærming da det krever bred erfaring med agile metoder (Campanelli og Parreiras, 2015). Det kan derfor argumenteres for at en annen metode kunne fungert bedre i visse sammenhenger, men som Conboy og Fitzgerald (2010) påpeker er en av de største utfordringene med contigency factors at det er ressurskrevende. Dette kan ses i sammenheng med den høye graden av formalitet og ressursbruk som en organisasjon krever for å kunne tilnærme seg method engineering i metodetilpasningen (Fitzgerald et al., 2000). Dette er ikke noe som har vært et stort problem i Telenor Digital, noe som sannsynligvis skyldes at det ikke var en dedikert metodeingeniør eller metodebase, og at det i stor grad var fokus på kontinuerlig forbedring. Det kan derfor tenkes at måten Telenor Digital har tilpasset agile metodikker har gjort dem bedre rustet til å håndtere usikkerhet, og dermed gjort dem mer agil.

5.2.3 To-delt tilnærming

Telenor Norge benytter seg av en litt annen tilnærming enn Telenor Digital. Der Telenor Digital til en viss grad implementerte en kjent agil metodikk, for så å drive kontinuerlig forbedring og tilpasning av den gjennom retrospektivmøter, viser funnene våre at Telenor Norge i større grad har konstruert metoden fra bunnen av. Når Telenor Norge tok avgjørelsen om å implementere agile metodikker i hele IT-divisjonen, ble dette gjort gjennom å implementere et rammeverk som la til rette for agile arbeidsmetodikker i stedet for å bare innføre en spesifikk agil metode i alle avdelinger. Dette ble altså gjort ved å innføre squads og tribes, for så å la squadene bestemme selv hvilke agile metodikker de vil benytte innad i squadene. Tilnærmingen for å utvikle squads-rammeverket kan sees på som det Conboy og Fitzgerald (2010) kaller method

engineering, men ikke på samme måte som i Telenor Digital. Empirien viser at det i Telenor Norge har vært et større fokus på å bygge opp et rammeverk fra bunnen for å gi teamene fleksibilitet. Resultatet ble det vi vil kalle et organiseringsrammeverk med svært få føringer for hvordan arbeidsmetodikken skulle benyttes på teamnivå. På teamnivå ble teamene, altså squadene, bedt om å benytte de agile metodikkene de selv mente var best, med andre ord kunne de benytte samme tilnærming som Telenor Digital, eller en helt annen tilnærming. I squadene som benytter agile metodikker i dag, har tilnærmingen i stor grad vært lik den i Telenor Digital, med en samling av metodefragment og retrospektiv for å tilpasse prosessene. Derfor mener vi at en tilnærming med å benytte squads legger til rette for kontinuerlig forbedring, et prinsipp som er viktig for Telenor Norge som helhet.

I enkelte squads har det vært veldig lite fokus på metoder og prosess, da utviklerne har kunnet jobbe uavhengig av hverandre og behovet for formaliserte prosesser ikke har vært like stort. Årsaken til dette mener vi kan knyttes opp mot tjenestene som squadene utviklet, da de representerer lav grad av kompleksitet og usikkerhet, og har dermed ikke har samme behov for formaliserte prosesser. Frem til nå hadde tjenester som ikke var vurdert som systemkritisk kunne blitt utviklet gjennom en squad, dette kan ses i sammenheng med de ikke-systemkritiske tjenestene i Telenor Digital som legger til rette for selvorganiserte team. Derimot for å sørge for at squadene tilnærmet seg den agile måten å jobbe på ble det bestemt at prosjektlederne skulle utdannes innen agil metodikk for å få en mer formell tilnærming. Fitzgerald et al. (2000) kaller en person som velger ulike metodefragment som skal brukes i hvert enkelt prosjekt for en metodeingeniør. Ved å heve kompetansen til prosjektlederne innen agil metodikk kan det tenkes at Telenor Norge skaper metodeingeniører som Fitzgerald et al. (2000) hevder må være tilstede for å lykkes med metodetilpasning. I en stor IT-organisasjon bestående av mange squads vil innføringen av agile metoder ifølge Nerur (2005) kreve lederskap og koordinering gjennom samarbeid. Ved å opprette metodeingeniører mener vi at Telenor Norge legger til rette for å håndtere den kaotiske prosessen som kritikerne hevder agile metoder står ovenfor. Disse vil dessuten kunne skape kunnskapsledelse som Dönmez og Grote (2018) mener må til for å kunne håndtere usikkerhet.

5.2.4 Formalitet og usikkerhet

Funnene relatert til tilnærming av metodetilpasning stemmer derfor i stor grad overens med funnene til Pedreira et al. (2007) om formalitet i metodetilpasning. I følge forskningen deres skjer metodetilpasning på en ganske uformell basis i små organisasjoner, mens de i større organisasjoner hadde en større tendens til å benytte seg av mer formaliserte prosesser. I Telenor Digital var prosessen forholdsvis uformell, men strukturert med tanke på retrospektivmøtene. I Telenor Norge ble metodetilpasningen på organisasjonsnivå gjennomført som en formell prosess, mens på teamnivå var derimot prosessen i Telenor Norge forholdsvis uformell. Dette kan tyde på at den friheten squadene fikk gjorde at de hadde mange av de samme forutsetningene for metodetilpasninger som mindre organisasjoner som Telenor Digital har. At squadene får frihet til å bestemme selv er noe vi mener er viktig, da det skaper større fleksibilitet for å kunne la squaden tilpasse metodikk til hvert enkelt prosjekt, og håndtere usikkerheten som oppstå

Grunnen til at metodetilpasningen i Telenor Digital og Telenor Norge skjedde på forskjellige måte mener vi i stor grad er på grunn av størrelsen på organisasjonene og arbeidsoppgavenes karakter. Telenor Digitals arbeidsoppgaver er ofte relativt like, og i forhold til Telenor Norge er de mindre avhengig av andre team for å gjennomføre oppgaver. Dette gjør det enklere å implementere agile rammeverk, da det er færre hensyn som må tas. Telenor Norge har også andre hensyn å ta enn Telenor Digital med tanke på usikkerhet og konsekvenser ved å gjøre feil. Dette er noe vi mener har vært førende for hvordan Telenor Norge har valgt å legge til rette for agile arbeidsmetodikker. Telenor Norges tilnærming med å implementere et organiseringsrammeverk som tilrettelegger for agile metoder, mener vi gjør det enklere for dem å implementere agile arbeidsmetodikker fremover, da det er enklere for squadene å vite hvilke hensyn som er relevante for akkurat dem, fremfor at ledelsen må ha kontroll på hele organisasjonen. Det at squadene har fått så stor grad av frihet tror vi dessuten skaper mer rom for kontinuerlig forbedring gjennom retrospektiver, og gjør det enklere for squadene å håndtere usikkerhet fortløpende. I tillegg legger squads-organiseringen til rette for at Telenor Norge enklere kan skifte mellom agil- og fossefallsmetodikk, og som intervjuobjektene påpeker kommer begge arbeidsmetodikkene til å være viktige i årene som kommer.

6 Konklusjon

Denne studien har hatt som formål å besvare problemstillingen «*Hva påvirker hvordan Telenor Norge og Telenor Digital tilnærmer seg agile metodikker for programvareutvikling?*». Dette har vi forsøkt å gjøre gjennom å intervjue utviklere, avdelingsledere og prosjektledere fra de to bedriftene. Konklusjonen av funnene og diskusjonen vår vil bli presentert i dette kapittelet.

Gjennom studien har det blitt klart at det er en del forskjeller i hvordan Telenor Norge og Telenor Digital tilnærmer seg bruk og tilpasning av agile metodikker. Der hvor Telenor Digital benytter utelukkende agile metodikker, har Telenor Norge valgt å legge til rette for en to-delt løsning, med agile metodikker i noen avdelinger og tradisjonell fossefallsmetodikk i andre. Tilpasningen av de agile metodikkene hos Telenor Digital har i stor grad skjedd gjennom bruk av retrospektivmøter, mens i Telenor Norge har man implementert et organiseringsrammeverk som legger til rette for bruk av agile metodikker. Denne løsningen legger til rette for at de selvorganiserte teamene selv kan bestemme hvordan de vil tilnærme seg tilpasning og bruk av agile metodikker.

Det som har vært førende for hvordan Telenor Norge og Telenor Digital tilnærmer seg agile metodikker er usikkerhet og deres tilnærmingen til den. Der hvor Telenor Norge har måtte fokusere på å minimere usikkerhet på grunn av konsekvensene ved å gjøre feil, har Telenor Digital fokusert på å håndtere den ved å skape mest mulig fleksibilitet, for eksempel gjennom agile praksiser som live-team. I Telenor Norge hvor mange av prosjektene har svært høy risiko, har de ikke hatt mulighet til å gå over til utelukkende agile metodikker selv om de har hatt et sterkt behov for det. Derfor har de implementert et organiseringsrammeverk som legger til rette for agile metodikker gjennom tverrfaglige, selvorganiserte team, samtidig som man kan benytte tradisjonell fossefallsmetodikk der hvor det anses som nødvendig. Denne tilnærmingen mener vi fører til at de kan beholde kontroll og minimere usikkerhet hvor det anses som nødvendig, samtidig som det skapes fleksibilitet til å håndtere usikkerhet der hvor konsekvensene ved å gjøre feil ikke er like store. Det kan tenkes at fremgangsmåten er nødvendig for å kunne tilpasse seg en bransje i konstant endring, hvor fleksibilitet og hurtighet er nøkkelfaktorer for å kunne holde seg konkurransedyktig, samtidig som man må ha kontrollmekanismer på plass for å unngå store feil.

Prosessen med å implementere og tilpasse agile metodikker har vært mer komplisert for Telenor Norge enn Telenor Digital. Dette mener vi i stor grad er på grunn av forskjellen på usikkerhet

og konsekvenser ved feil i de to bedriftene, men også på grunn av avhengigheter til andre team og generelt størrelsen på bedriftene. Ved å implementere et organiseringsrammeverk som fokuserer på selvorganiserte team, ser vi at teamene i Telenor Norge får mange av de samme egenskapene som teamene i Telenor Digital har, og dermed klarer å benytte og tilpasse agile metodikker på en lignende måte. På samme måte som man i agile metodikker deler opp oppgaver i små og håndterlige biter for å gjennomføre dem kjapt, har Telenor Norge delt opp avdelingene i små og håndterlige team slik at de kan arbeide kjapt – og agilt.

6.1 Implikasjoner

Det teoretiske grunnlaget for agile metodikker kan virke banebrytende og noe idealistisk sammenlignet med tradisjonell metodikk, det er derfor interessant å se hvordan dette brukes og tilpasses i praksis, og hva som er førende for prosessene bak. I dette tilfellet ble tilnærming til usikkerhet og risiko sett på som de viktigste faktorene. Denne studien sier noe om hvordan Telenor Norge og Telenor Digital har tilnærmet seg disse faktorene, men gir ikke noe konkret svar på hva som er den optimale løsningen. Om dette er å begrense usikkerhet ved å dele opp prosjektet eller å benytte andre metodikker som i større grad tar høyde for det er uklart, men det er tydelig at det er et behov for å håndtere usikkerhet og risiko i agile prosjekter. For Telenor Norge kan det derfor være nyttig å undersøke hvordan de kan få til å jobbe agilt på prosjekter med høy risiko. Videre er det tydelig at Telenor Digital burde se nærmere på hvordan de kan legge til rette for bedre læring gjennom retrospektivene, da det i stor grad er enkelkretslæring som blir resultatet i dag. For andre bedrifter, kan det være nyttig å se å bruke denne studien for å lære av hvordan Telenor Norge og Telenor Digital har tilnærmet seg agile metodikker. Siden vi i denne studien både har studert en «stor» og en «liten» bedrift, tror vi studien kan være nyttig for de fleste bedrifter som driver med programvareutvikling.

6.2 Videre forskning

Til videre forskning vil vi anbefale å studere driverne til tilnærming av bruk og tilpasning av agile metodikker nærmere. I denne studien ble håndtering av usikkerhet funnet som den viktigste faktoren, men er dette alltid tilfellet? Det vil også være interessant å se på nøyaktig hva man kan gjøre for å legge til rette for agile arbeidsmetodikker i bedrifter hvor håndtering av usikkerhet er den viktigste faktoren. I denne casen gjorde Telenor Norge det ved å benytte fossefallsmetodikk hvor konsekvensene ved å gjøre feil ble vurdert som høy, men er det mulig å legge til rette for agil arbeidsmetodikk også i disse tilfellene? Dette er noe vi mener kan være

interessant å forske nærmere på, da det vil ha stor praktisk betydning for mange bedrifter, samt at det finnes lite forskning på det pr dags dato.

Vi mener også det vil være interessant å gå mer i dybden på retrospektivmøtene som arena for metodetilpasning, og se på for eksempel hvordan man kan legge til rette for dobbelkretslæring og større tilpasninger av arbeidsmetodikk.

Aakre, J. & Scharning, H. S. 2013. *Prosjekthåndboka 2.0: verktøykasse for kreative team*, Universitetsforlaget.

Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. 2002. *Agile Software Development Methods: Review and Analysis*.

Alliance, S. 2016. Learn about scrum. *Scrum Alliance*.

Ammouri, Y. M. 2015. *Agile software development, Basics and fundamentals* [Online]. Codeproject. Available: https://www.codeproject.com/Articles/1064114/Agile-Software-Development-Basics [Accessed].

Atkinson, R. 1999. Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International journal of project management,* 17**,** 337-342.

Augustine, S. 2005. *Managing agile projects*, Prentice Hall PTR.

Aydin, M. N., Harmsen, F., Van Slooten, K. & STEGWEE, R. 2004. An agile information systems development method in use. *Turkish Journal of Electrical Engineering & Computer Sciences,* 12**,** 127-138.

Banijamali, A., Dawadi, R., Ahmad, M. O., Similä, J., Oivo, M. & Liukkunen, K. 2017. *Empirical Investigation of Scrumban in Global Software Development*.

Bannerman, P. L. 2008. Risk and risk management in software projects: A reassessment. *Journal of Systems and Software,* 81**,** 2118-2133.

Brooks, F. P. J. 1987. No Silver Bullet Essence and Accidents of Software Engineering. *Computer,* 20**,** 10-19.

Campanelli, A. S. & Parreiras, F. S. 2015. Agile methods tailoring – A systematic literature review. *Journal of Systems and Software,* 110**,** 85-100.

Cockburn, A. 2002. *Agile software development*, Addison-Wesley longman publishing Co.

Cohen, D. L., Mikael; Costa, Patricia; 2004. An introduction to agile methods. *Advances in computers,* 62**,** 1-66.

Collier, B., DeMarco, T. & Fearey, P. 1996. A defined process for project post mortem review. *IEEE software,* 13**,** 65-72.

Collins, G. 2017. Appendix 1 - Agile Project Management A2 - Lester, Eur Ing Albert. *Project Management, Planning and Control (Seventh Edition).* Butterworth-Heinemann.

Conboy, K. 2009. *Agility From First Principles: Reconstructing the Concept of Agility in Information Systems Development*.

Conboy, K. & Fitzgerald, B. 2010. Method and developer characteristics for effective agile method tailoring: A study of xp expert opinion. *ACM Transactions on Software Engineering and Methodology,* 20.

Creswell, J. W. 2014. *Research design : qualitative, quantitative, and mixed methods approaches,* Los Angeles, Calif, SAGE.

Diallo, U. & Patel, P. 2014. The Digital Imperative.

Difi. 2018. *Prosjektveiviseren* [Online]. Available: https://www.prosjektveiviseren.no/ [Accessed].

Dingsøyr, T., Nerur, S., Balijepally, V. & Brede Moe, N. 2012. A decade of agile methologies: Towards explaining agile software development. *Journal of systems and software,* 85**,** 1213-1221.

Dybå, T. D., Torgeir; 2009. What do we know about Agile software development. *IEEE Software,* 26**,** 6-9.

Dönmez, D. & Grote, G. 2018. Two sides of the same coin–how agile software development teams approach uncertainty as threats and opportunities. *Information and Software Technology,* 93**,** 94-111.

Fitzgerald, B., Hartnett, G. & Conboy, K. 2006. *Customising agile methods to software practices at Intel Shannon*.

Fitzgerald, B., Russo, N. & O'Kane, T. 2000. *An Empirical Study of System Development Method Tailoring*.

Fontana, R. M., Meyer Jr, V., Reinehr, S. & Malucelli, A. 2015. Progressive Outcomes: A framework for maturing in agile software development. *Journal of Systems and Software,* 102**,** 88-108.

Forbes, L. H. & Ahmed, S. M. 2010. *Modern construction: lean project delivery and integrated practices*, Crc Press.

Fowler, M. & Highsmith, J. 2001. The agile manifesto. *Software development,* 9**,** 28-35.

Friedman, A. L. & Miles, S. 2006. *Stakeholders: Theory and practice*, Oxford University Press on Demand.

G. Guba, E. & S. Lincoln, Y. 2018. *Fourth generation evaluation / Egon G. Guba, Yvonna S. Lincoln*.

Gartner. 2014. *Gartner Says in the Digital World CIOs Need Bimodal IT: Rock Solid IT With*

Ability for Fluidity [Online]. Available: https://www.gartner.com/newsroom/id/2865718 [Accessed 2018].

Gregory, P., Barroca, L., Sharp, H., Deshpande, A. & Taylor, K. 2016. The challenges that challenge: Engaging with agile practitioners' concerns. *Information and Software Technology,* 77**,** 92-104.

Gregory, P., Barroca, L., Taylor, K., Salah, D. & Sharp, H. Agile challenges in practice: a thematic analysis. International Conference on Agile Software Development, 2015. Springer, 64-80.

Grote, G. 2009. *Management of uncertainty: Theory and application in the design of systems and organizations*, Springer Science & Business Media.

Haffke, I., Kalgovas, B. & Benlian, A. 2017. Options for Transforming the IT Function Using Bimodal IT. *MIS Quarterly Executive,* 16.

Henderson-Sellers, B., Ralyté, J., Ågerfalk, P. J. & Rossi, M. 2014. *Situational Method Engineering*, Springer Berlin Heidelberg.

Henderson-Sellers, B. S., Magdy; 2005. Creating a Dual-Agility Method: The Value of Method Engineering. *Journal of Database Management,* 16**,** 1-23.

Highsmith, J. 2003. Agile project management: Principles and tools. Agile Project Management. *Cutter Consortium.* Cutter.

Highsmith, J. & Cockburn, A. 2001. Agile software Development: The business of innovation. *Computer,* 34**,** 120-127.

Horlach, B., Drews, P. & Schirmer, I. 2016. Bimodal IT: Business-IT alignment in the age of digital transformation. *Multikonferenz Wirtschaftsinformatik (MKWI)***,** 1417-1428.

Ibrahim, H., Far, B. H., Eberlein, A. & Daradkeh, Y. Uncertainty management in software engineering: Past, present, and future. Electrical and Computer Engineering, 2009. CCECE'09. Canadian Conference on, 2009. IEEE, 7-12.

Iivari, J. 1989. *A methodology for IS development as organizational change: A pragmatic contingency approach*.

Ikonen, M., Kettunen, P., Oza, N. & Abrahamsson, P. Exploring the Sources of Waste in Kanban Software Development Projects. 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 1-3 Sept. 2010 2010. 376-381.

Jalonen, H. 2011. The uncertainty of innovation: a systematic review of the literature. *Journal of Management Research,* 4.

Javdani Gandomani, T., Zulzalil, H., abdul ghani, A. a., Md Sultan, A. B. & Ziaei, M. 2013. *Obstacles in moving to agile software development methods; At a Glance*.

Johannessen, A., Christoffersen, L. & Tufte, P. A. 2011. Forskningsmetode for økonomisk-administrative fag. 490.

Jørgensen, M. 2016. A survey on the characteristics of projects with success in delivering client benefits. *Information and software technology,* 78**,** 83-94.

Kniberg, H. & Skarin, M. 2010. *Kanban and Scrum: making the most of both,* USA, C4Media.

Knight, F. H. 2012. *Risk, uncertainty and profit*, Courier Corporation.

Kolltveit, B. J., Karlsen, J. T. & Grønhaug, K. 2004. Exploiting opportunities in uncertainty during the early project phase. *Journal of Management in Engineering,* 20**,** 134-140.

Koskela, J. & Abrahamsson, P. On-site customer in an XP project: Empirical results from a case study. European Conference on Software Process Improvement, 2004. Springer, 1-11.

Larman, C. 2004. *Agile and iterative development: a manager's guide*, Addison-Wesley Professional.

Lyytinen, K. & Robey, D. 1999. Learning failure in information systems development. *Information Systems Journal,* 9**,** 85-101.

Maruping, L. M., Venkatesh, V. & Agarwal, R. 2009. A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research,* 20**,** 377-399.

Moe, N. B., Aurum, A. & Dybå, T. 2012. Challenges of shared decision-making: A multiple case study of agile software development. *Information and Software Technology,* 54**,** 853-865.

Moe, N. B., Dingsøyr, T. & Dybå, T. 2008. Overcoming barriers to self-management in software teams. *IEEE Software*.

Morgan, G. 2004. *Organisasjonsbilder: innføring i organisasjonsteori*, Gyldendal Akademisk.

Myllyaho, M., Salo, O., Kääriäinen, J., Hyysalo, J. & Koskela, J. 2004. A review of small and large post-mortem analysis methods. *Proceedings of the ICSSEA, Paris***,** 1-8.

Nelson, D. 1980. *Frederick W. Taylor and the rise of scientific management*, Madison: University of Wisconsin Press.

Nelson, R. R. IT Project Retrospectives: Learning from the Past through a Program of Action Research. AMCIS, 2010. 475.

Nerur, S. P. M., Radhakanta; Mangalaraj, George; 2005. Challenges of migrating to agile methodologies. *Communications of ACM,* 48**,** 72-78.

Owen, R., Koskela, L., Henrich, G. & Codinhoto, R. Is agile project management applicable to construction? Proceedings of the 14th Annual Conference of the International Group for Lean Construction, 2006. 51-66.

Pedreira, O., Piattini, M., Luaces, M. R. & Brisaboa, N. R. 2007. A systematic review of software process tailoring. *SIGSOFT Softw. Eng. Notes,* 32**,** 1-6.

Penrose, E. T. 2009. *The Theory of the Growth of the Firm*, Oxford university press.

Petersen, K., Wohlin, C. & Baca, D. 2009. the waterfall model in large svale development. *PROFES. Springer***,** 386-400.

Rodríguez, P., Markkula, J., Oivo, M. & Turula, K. 2012. Survey on agile and lean usage in finnish software industry. *Proceedings of the ACM-IEEE international symposium on Empirical Software Engineering and Measurement. IEEE***,** 139-148.

Rolfsen, M. 2000. *Trendenes tyranni: produksjon og arbeid i et nytt rhundre*, Fagbokforl.

Rolland, C. & Prakash, N. 1996. A Proposal For Context-Specific Method Engineering. *In:* BRINKKEMPER, S., LYYTINEN, K. & WELKE, R. J. (eds.) *Method Engineering: Principles of method construction and tool support.* Boston, MA: Springer US.

Rolland, K. H., Mikkelsen, V. & Næss, A. Tailoring Agile in the Large: Experience and Reflections from a Large-Scale Agile Software Development Project. *In:* SHARP, H. & HALL, T., eds. Agile Processes, in Software Engineering, and Extreme Programming, 2016// 2016 Cham. Springer International Publishing, 244-251.

Rolstadås, A. 2006. *Praktisk prosjektstyring*, Tapir akademisk forlag.

Rose, K. H. 2013. A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition. *Project management journal,* 44.

Royce, W. W. 1987. Managing the development of large software systems: concepts and techniques. *Proceedings of the 9th international conference on software engineering IEEE Computer society press*, 328-338.

Schartum, D. W. & Sørensen, D. 2014. CompLex.

Schwaber, K. & Sutherland, J. 2017. *Scrum framework* [Online]. Available: https://www.scrum.org/resources/scrum-framework-poster [Accessed].

Seyam, M. 2010. *Agile Methodologies in Information Systems Development: How to be Agile, without losing the disciplines of being Traditional*, LAP Lambert Academic Publishing.

Singh, N. P. & Soni, R. 2011. Agile software: Ensuring quality assurance and processes. *High Performance Architecture and Grid Computing.* Springer.

Strand, K. & Karlsen, K. 2014. Agile Contracting and Execution. PROMIS.

Sutherland, J. & Schwaber, K. 2017. *The scrum guide* [Online]. Available: http://www.scrumguides.org/scrum-guide.html [Accessed].

Telenor. 2018. *Bakgrunn og strategi* [Online]. Available: https://www.telenor.com/no/om-oss/strategi/ [Accessed].

Tjora, A. H. 2017. *Kvalitative forskningsmetoder i praksis,* Oslo, Gyldendal akademisk.

Turk, D., France, R. & Rumpe, B. 2002. Limitations of agile software processes. *XP2002*, 43-46.

Versionone. 2017. *11th Annual state of agile report* [Online]. Version one. Available: https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2 [Accessed].

Vinekar, V. S., Craig W.; Nerur, Sridhar; 2006. Can agile and traditional systems development approaches coexist? An ambidextrous view. *Information systems management,* 23, 31-42.

Williams, L. & Cockburn, A. 2003. Agile software development: it's about feedback and change. *Computer,* 36, 39-43.

Yin, R. K. 2009. *Case Study Research: Design and Methods*, SAGE Publications.

Zhiying, Z. 2003. CMM in uncertain environments. *Commun. ACM,* 46, 115-119.

7 Vedlegg

7.1 Vedlegg 1: intervjuguide

Forklaring
Spørsmål og ting vi skal si/informere om er market med "-".
Innrykket tekst i kursiv er ment som eventuelle ting å stille oppfølgningsspørsmål om hvis informant ikke nevner det i overstående spørsmål.

Innledning
Formål: Informere om praktiske detaljer rundt prosjektet og intervjuet
-Presentere oss
-Informere om prosjektet og hva vi skal spørre om
-Informere om hvordan data behandles
-Informere om anonymitet
-Informere om mulighet til å avbryte intervju
-Indikere lengde på intervju
-Opptak
-Noen spørsmål så langt? Har du noen spørsmål underveis får du bare gi beskjed.

Faktaspørsmål
Formål: Etablere tillit og "varme opp" informanten for videre intervju
-Hvor lenge har du jobbet i Telenor?
-Hva er din stilling i Telenor?
-Har du hatt noen andre stillinger i Telenor?
-Hvilken utdanning har du?

Introduksjonsspørsmål (generelt)
Formål: Rette informanten mot temaet smidig arbeidsmetodikk og metodetilpasning
"Denne oppgaven handler som nevnt om smidig arbeidsmetodikk og hvordan arbeidsmetodikker utvikles her i Telenor"

-Hva tenker du når du hører smidig/agil arbeidsmetodikk?
-Hva tenker du når du hører metodetilpasning?

Hoveddel (spesifikt)
Formål: Få innsikt i hvordan informanten oppfatter arbeidsmetodikk i sin og andre avdelinger, hvorfor arbeidsmetodikken er slik den er og hvordan den har utviklet seg til å bli slik.
-Kan du si litt om hvordan dere jobber i din avdeling i dag?
-Teamstørrelse: hvordan organiserer dere størrelse, flere team eller et dedikert?
-ulike roller i team
-Faste ting som gjøres hver dag? (Standup o.l.)
-Er det noen utfordringer med hvordan dere jobber?
-Faser? Oppdeling av arbeidsoppgaver?
-Ofte leveranser?
-Jobber dere etter noen navngitte metodikker/prosjektmodell? Om ikke, hvordan kan dere enes om arbeidsmetodikk
-Hva kjennetegner den metodikken?

-Hvis bestemt arbeidsmetodikk: Avviker dere fra den bestemte metodikken?

-Hvis arbeidsmetodikk varierer: hva bestemmer type arbeidsmetodikk?
-Prosjektstørrelse/teamstørrelse?

-Leveransefrist?
-Produkttype?

-Kan du si litt om din rolle i dette?
-Agil jobbing
-Tilpasning av metode
-Kommunikasjon av metodikk til ansatte

-Bruker dere andre arbeidsmetodikker?
- Har isåfall disse blitt endret/tilpasset underveis?

-Bytter dere ofte arbeidsmetodikk?
-hvorfor?
-Hvorfor jobber dere akkurat slik?
-Bestemt "ovenfra"?
-Utviklet seg naturlig?

-Hvordan er arbeidsmetodikken kommunisert?

-Jobbes det på andre måter i andre avdelinger, vet du isåfall hvorfor?
-Ulike typer prosjekt?
-
-Tenker dere noe over hvordan dere jobber?
-Utviklerninvå, prosjektleder, osv
-Videreutvikling av metodikk/optimering
-Er det noe dere kunne gjort annerledes?

Avslutning

-Før vi avslutter, er det noe mer du tenker kan være relevant for å avdekke arbeidsmetodikk og metodetilpasning i Telenor?
-Takk for at du tok deg tid til intervjuet.

7.2 Vedlegg 2: Godkjent forenklet vurdering, NSD

Frode Heldal

7491 TRONDHEIM

Vår dato: 23.01.2018 Vår ref: 58347 / 3 / STM Deres dato: Deres ref:

Forenklet vurdering fra NSD Personvernombudet for forskning

Vi viser til melding om behandling av personopplysninger, mottatt 12.01.2018.
Meldingen gjelder prosjektet:

58347	*Hvordan kan man oppnå måloppnåelse ved bruk av autonomi i en kunnskapsbedrift?*
Behandlingsansvarlig	*NTNU, ved institusjonens øverste leder*
Daglig ansvarlig	*Frode Heldal*
Student	*Viktor Myrvang*

Vurdering

Etter gjennomgang av opplysningene i meldeskjemaet med vedlegg, vurderer vi at prosjektet er omfattet av personopplysningsloven § 31. Personopplysningene som blir samlet inn er ikke sensitive, prosjektet er samtykkebasert og har lav personvernulempe. Prosjektet har derfor fått en forenklet vurdering. Du kan gå i gang med prosjektet. Du har selvstendig ansvar for å følge vilkårene under og sette deg inn i veiledningen i dette brevet.

Vilkår for vår vurdering

Vår anbefaling forutsetter at du gjennomfører prosjektet i tråd med:

- opplysningene gitt i meldeskjemaet
- krav til informert samtykke
- at du ikke innhenter sensitive opplysninger
- veiledning i dette brevet
- NTNU sine retningslinjer for datasikkerhet

Veiledning

Krav til informert samtykke

Utvalget skal få skriftlig og/eller muntlig informasjon om prosjektet og samtykke til deltakelse.
Informasjon må minst omfatte:

- at NTNU er behandlingsansvarlig institusjon for prosjektet
- daglig ansvarlig (eventuelt student og veileders) sine kontaktopplysninger
- prosjektets formål og hva opplysningene skal brukes til

Dokumentet er elektronisk produsert og godkjent ved NSDs rutiner for elektronisk godkjenning.

NSD – Norsk senter for forskningsdata AS Harald Hårfagres gate 29 Tel: +47-55 58 21 17 nsd@nsd.no Org.nr. 985 321 884
NSD – Norwegian Centre for Research Data NO-5007 Bergen, NORWAY Faks: +47-55 58 96 50 www.nsd.no

- hvilke opplysninger som skal innhentes og hvordan opplysningene innhentes
- når prosjektet skal avsluttes og når personopplysningene skal anonymiseres/slettes

På nettsidene våre finner du mer informasjon og en veiledende mal for informasjonsskriv.

Forskningsetiske retningslinjer
Sett deg inn i forskningsetiske retningslinjer.

Meld fra hvis du gjør vesentlige endringer i prosjektet
Dersom prosjektet endrer seg, kan det være nødvendig å sende inn endringsmelding. På våre nettsider finner du svar på hvilke endringer du må melde, samt endringsskjema.

Opplysninger om prosjektet blir lagt ut på våre nettsider og i Meldingsarkivet
Vi har lagt ut opplysninger om prosjektet på nettsidene våre. Alle våre institusjoner har også tilgang til egne prosjekter i Meldingsarkivet.

Vi tar kontakt om status for behandling av personopplysninger ved prosjektslutt
Ved prosjektslutt 25.05.2018 vil vi ta kontakt for å avklare status for behandlingen av personopplysninger.

Gjelder dette ditt prosjekt?

Dersom du skal bruke databehandler
Dersom du skal bruke databehandler (ekstern transkriberingsassistent/spørreskjemaleverandør) må du inngå en databehandleravtale med vedkommende. For råd om hva databehandleravtalen bør inneholde, se Datatilsynets veileder.

Hvis utvalget har taushetsplikt
Vi minner om at noen grupper (f.eks. opplærings- og helsepersonell/forvaltningsansatte) har taushetsplikt. De kan derfor ikke gi deg identifiserende opplysninger om andre, med mindre de får samtykke fra den det gjelder.

Dersom du forsker på egen arbeidsplass
Vi minner om at når du forsker på egen arbeidsplass må du være bevisst din dobbeltrolle som både forsker og ansatt. Ved rekruttering er det spesielt viktig at forespørsel rettes på en slik måte at frivilligheten ved deltakelse ivaretas.

Se våre nettsider eller ta kontakt med oss dersom du har spørsmål. Vi ønsker lykke til med prosjektet!

Vennlig hilsen

Marianne Høgetveit Myhren

Siri Tenden Myklebust

Kontaktperson: Siri Tenden Myklebust tlf: 55 58 22 68 / Siri.Myklebust@nsd.no