

Anne Sofie Nagelhus

Brukermedvirkning i smidig utvikling av IT-produkter

Et casestudium om utvikling av løsning for sykepengesaksbehandling i NAV

Masteroppgave i Produktutvikling og produksjon

Veileder: Bjørn Andersen

Juni 2022

Anne Sofie Nagelhus

Brukermedvirkning i smidig utvikling av IT-produkter

Et casestudium om utvikling av løsning for
sykepengesaksbehandling i NAV

Masteroppgave i Produktutvikling og produksjon
Veileder: Bjørn Andersen
Juni 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for ingeniørvitenskap
Institutt for maskinteknikk og produksjon



Kunnskap for en bedre verden

Forord

Denne masteroppgaven gjennomføres gjennom Norges Tekniske og Naturvitenskapelige Universitet, NTNU. Emnet er under institutt for maskinteknikk og produksjon, MTP. Masteroppgaven gjennomføres over 20 uker i løpet av våren 2022 og teller 30 studiepoeng i TPK4920 – Prosjekt- og kvalitetsledelse. Oppgaven utføres i samarbeid med NAV, og observasjonene er fra et utviklingsteam hos dem.

Gjennom arbeidet med denne oppgaven har jeg fått et innblikk i hvordan brukermedvirkningsarbeid kan gjennomføres i utvikling av IT-systemer. Det har vært veldig spennende og lærerikt.

Jeg har satt noen mål for masteroppgaven for å sette den i en større sammenheng. Resultatmålene er å kartlegge hvordan et av utviklingsteamene til NAV jobber med brukermedvirkningsaktiviteter og å finne forbedringsmuligheter eller bekrefte at arbeidsmetoden er god nok. Dette anses som nådd gjennom de fire siste kapitlene i oppgaven. Effektmålet er at NAV tar lærdom av funnene gjennom masteroppgaven og at brukermedvirkning får den plassen det trenger i utviklingsprosjekter hos NAV. Samfunnsålet til masteroppgaven er å tilføye mer lærdom rundt tematikken *brukermedvirkning* og at andre utviklingsteam kan lære av erfaringene til teamet fra dette studiet. Effektmålet og samfunnsålet kan det ikke konkluderes om er nådd gjennom denne oppgaven.

Jeg ønsker å takke veilederen min, Bjørn Andersen, for god veiledning og oppfølging angående oppgavens fremgangsmåte, omfang og oppsett. Videre vil jeg takke Nina Bakken og Solveig Grieg for samarbeidet og at de gjorde det mulig å gjennomføre denne masteroppgaven for NAV. Jeg vil også takke alle som stilte til intervju og som var med på observasjonsdager.

Til slutt ønsker jeg å takke familie, samboer, venner og studiekamerater for støtte gjennom mastersemesteret og for å bidra til en veldig fin studietid.

Trondheim

Juni 2022

Anne Sofie Nagelhus

Abstract

[Theme] This master's thesis addresses user involvement in agile development of an IT-system for case processing of sick leave cases. **[Purpose]** The purpose is to investigate how a development team in NAV which is using agile methods can improve their user involvement processes. **[Method]** There has been completed a case study of a development team through a literature study, observations, and interviews. **[Results]** The caseworkers, who are the users of the system, are involved continuously by communicating through the communication platform, Slack, any time. Team members believe that there is a low threshold for posting questions and remarks on Slack, while some caseworkers deem the threshold for posting on Slack can be high. There are weekly meetings where team members observe caseworkers processing cases in the new system. Both caseworkers and team members say that they feel it is difficult to communicate to the other part why something is considered to be prioritized as it is. A part of the user involvement activities seems to depend on individuals. Much of the user involvement depend on engaged users; the entire user group may therefore not be represented in the user involvement process. Some of the caseworkers are called "super-caseworkers"; these are the caseworkers who have the most knowledge of the system. "Normal" caseworkers must via the super-caseworkers to share detected faults and flaws of the system. It is important for the team that both parties trust each other; caseworkers express that they are taken seriously by the team members, which may increase the trust. **[Conclusion]** The organizing of the user involvement activities should be changed so it is not dependent on individuals to be executed. The time from team members have learned about new demands from users to it is implemented should be reduced so the team is able to take advantage of involving the users continuously combined with agile development. It is important that the management above the team makes a knowingly decision regarding who evaluates the importance of the issues addressed by the users; is it the super-caseworkers, the development team members, or the users themselves.

Sammendrag

[Tema] Denne masteroppgaven tar for seg brukermedvirkning i smidig utvikling av et IT-system for behandling av sykepengesaker. **[Hensikt]** Hensikten er å undersøke hvordan et utviklingsteam som benytter smidige metoder i NAV kan bli bedre på sine brukermedvirkningsprosesser. **[Metode]** Det har blitt gjennomført et casestudium av et utviklingsteam gjennom litteraturstudium, observasjon og intervjuer. **[Resultater]** Saksbehandlerne, som er brukerne av systemet, involveres kontinuerlig ved at de kan ta kontakt via kommunikasjonsplattformen, Slack, når som helst. Teammedlemmer mener at det er lav terskel for å legge ut spørsmål og bemerkninger på Slack, mens noen saksbehandlere mener det kan være høy terskel å legge ut ting de lurte på der. Det arrangeres ukentlige saksbehandlertreff der teammedlemmer observerer at saksbehandlere behandler saker i det nye systemet. Både saksbehandlere og teammedlemmer sier de synes det er vanskelig å kommunisere til den andre parten hvorfor noe er viktig og hvorfor noe prioriteres slik det blir prioritert. En del av brukermedvirkningsaktivitetene anses å avhenge av enkeltpersoner. Mye av brukermedvirkningen avhenger av engasjement fra brukerne, noe som kan føre til at hele brukergruppa ikke blir representert i medvirkningsprosessen. De av saksbehandlerne som har best kjennskap til systemet kalles supersaksebehandlere. «Vanlige» saksbehandlere må via dem for å dele oppdagede feil og mangler ved systemet. Det er viktig for teamet at begge parter har tillit til hverandre; saksbehandlere uttrykker at teammedlemmene tar dem seriøst og at de er flinke til å sette seg inne saksbehandlerne sin situasjon, noe som kan bidra til økt tillit. **[Konklusjon]** Organiseringen av brukermedvirkningsaktiviteter bør endres slik at de ikke avhenger av enkeltpersoner for å bli gjennomført. Tiden fra teammedlemmer har lært om nye behov fra brukerne til det blir implementert i systemet bør forkortes slik at teamet får utnyttet kontinuerlig brukerinvolvering i kombinasjon med smidig utvikling. Det er viktig at ledelsen over teamet gjør et bevisst valg om hvem som skal vurdere viktigheten av problemstillinger som tas opp fra brukerne; er det supersaksbehandlerne, utviklingsteammedlemmer eller brukerne selv.

Innhold

Forord	v
Abstract	vi
Sammendrag	vii
Figurer	xi
Tabeller	xii
1 Introduksjon	1
1.1 Problemstilling og forskningsspørsmål	1
1.2 Bakgrunn	1
1.3 Studiets begrensninger	1
1.4 Oppgavens oppbygging	2
2 Metode	3
2.1 Forskningsdesign	3
2.2 Casestudium	4
2.3 Litteraturstudium	4
2.4 Observasjon	5
2.5 Intervjuer	6
2.6 Analyse	6
2.7 Evaluering av forskningsdesign	8
2.7.1 Evaluering av casestudium	8
2.7.2 Evaluering av litteraturstudium	9
2.7.3 Evaluering av observasjon	9
2.7.4 Evaluering av intervjuer	10
3 Teori	12
3.1 Brukermedvirkning	12
3.1.1 Definisjoner og viktigheten av brukermedvirkning	12
3.1.2 Når bør brukerne involveres?	12
3.1.3 Brukermedvirkningsmodell	13
3.1.4 Utvelgelse av brukere til medvirkningsprosessen	14
3.1.5 Behov- og kravspesifikasjoner	14
3.1.6 Testing og innhenting av data om brukeropplevelse	15
3.2 Arbeidsmetoder	17
3.2.1 Key performance indicators (KPI)	17

3.2.2	Forskjellige planleggingsmetoder	17
3.2.3	Prosjekt	17
3.2.4	Fossefallsmetoden	18
3.2.5	Smidig (agil) arbeidsmetode	19
3.2.6	Kontinuerlig eksperimentering	20
3.2.7	Produktbasert utvikling	20
3.3	Team- og organisasjonsmodeller	21
3.3.1	Spotifymodellen	21
3.3.2	Extreme programming (XP)	23
4	NAV som caseorganisasjon	24
4.1	NAV som organisasjon	24
4.2	Proessen for søknad om sykepenger	24
4.3	Utviklingsteamets oppbygging og ansvarsområder	25
4.4	Utviklingsteamets arbeidsmetoder	27
4.5	Produktet	30
5	Resultater	32
5.1	Teamet og organisering	34
5.2	Holdninger og motivasjon til brukermedvirkningsaktiviteter	36
5.3	Kommunikasjon	37
5.4	Dokumentasjon	41
5.5	Supersaksbehandlerrollen	41
5.6	Saksbehandlertreff	42
5.7	Finansiering av utviklingen	45
6	Diskusjon	46
6.1	Organisering	46
6.2	Teamets arbeidsmetoder	47
6.3	Kommunikasjon rundt felles mål, visjoner og arbeidsmetode	48
6.4	Brukermedvirkning i startprosessen	50
6.5	Organisering av brukermedvirkningen	50
6.5.1	Generell brukermedvirkning	50
6.5.2	Om supersaksbehandlerrollen	52
6.5.3	Saksbehandlertreff	53
6.6	Teamets lærdom gjennom brukermedvirkning	54

6.7	Terskel for å legge ut på Slack.....	56
6.8	Dokumentasjonskanaler.....	57
7	Konklusjon og videre arbeid	59
7.1	Konklusjon	59
7.1.1	Hvordan anbefaler litteraturen at brukerne skal involveres i IT-utvikling?.....	59
7.1.2	Hvordan involveres brukerne i utviklingen av systemet for behandling av sykepengesaker?	59
7.1.3	Hvilke brukermedvirkningsaktiviteter bør endres?	60
7.2	Videre forskning	62
	Kilder	63
	Vedlegg	70
A.	Observasjonsplan.....	70
B.	Intervjuguide generelt om brukermedvirkningsprosesser	71
C.	Intervjuguide spesifikt til observasjonsaktiviteten	73
D.	Det smidige manifest.....	75
E.	De syv prinsippene for brukersentrert smidig utvikling i NAV	77

Figurer

Figur 3.1: Grunnleggende struktur for brukersentrert smidig utvikling (Beyer, 2010, p. 27)	14
Figur 3.2: Mengden defekter som oppstår ved sjelden testing (øverst) versus hyppig testing (nederst) (Beck & Andres, 2005, p. 99)	16
Figur 3.3: «Forskjellige planleggingsmodeller» (Andersen, 2018, p. 213).	17
Figur 3.4: Visualisering av Samset sin faseinndeling av et prosjekt (Samset, 2015, pp. 46–47).....	18
Figur 3.5: Fossefallmodell basert på Andersen (2018, p. 22) sin modell	19
Figur 3.6: Smidig arbeidsmetode (Adam, 2021; Admin, 2020; Introduction to Agile Methodology, 2020).	19
Figur 3.7: Norsk versjon av produkt-marked-tilpasningspyramiden (Olsen, 2015, p. xix). .	21
Figur 3.8: Spotifymodellen (Kniberg & Ivarsson, 2012, p. 1).	22
Figur 4.1: Teamets oppbygging.....	26
Figur 4.2: Parter som teamet påvirker og som påvirker teamet.....	30
Figur 4.3: Forenklet visualisering av produktet som leveres fra utviklingsteamet.....	31

Tabeller

Tabell 2.1: Koder fra observasjon.....	7
Tabell 2.2: Koder fra intervjuene.....	8
Tabell 5.1: Oppsummering av funn.....	34

1 Introduksjon

I dette kapittelet introduseres problemstilling og forskningsspørsmål. Deretter presenteres bakgrunn for valgt tema, etterfulgt av studiets begrensninger og oppgavens oppbygging.

1.1 Problemstilling og forskningsspørsmål

Problemstillingen i denne oppgaven er:

«Hvordan involveres brukerne i produktutviklingen av systemet for behandling av sykepengesaker i NAV sammenlignet med anbefalinger fra relevant litteratur, og hvordan kan brukermedvirkningsprosessen forbedres?»

Forskningsspørsmålene som ønskes besvart er:

1. Hvordan anbefaler litteraturen at brukerne skal involveres i IT-utvikling?
2. Hvordan involveres brukerne i utviklingen av systemet for behandling av sykepengesaker?
3. Hvilke brukermedvirkningsaktiviteter bør endres?

1.2 Bakgrunn

Robertson og Robertson (2013, p. xxii) sier at siden software er en løsning, så må man forstå det virkelige problemet før man kan løse det. Brukernes behov fra systemet er fundamentet til løsningen i de fleste IT-løsninger. Når man benytter smidige metoder trenger man kontinuerlig kommunikasjon med brukerne av systemet (Lamelas, 2018).

«Det finnes en stor mengde modeller for smidige metoder, men det er lite i arbeidsformene som er forskningsbasert» (Eriksen, 2022). Brukermedvirkning i utviklingsprosjekter som benytter smidig metode er et tema som fasinere meg. Derfor ønsket jeg å studere et utviklingsteam i NAV for å lære hvordan de gjennomfører brukermedvirkningsaktiviteter og om de har forbedringsmuligheter på dette punktet. Teamet jobber produktbasert, som vil si at produktets leveranse er i fokus gjennom utviklingen, uten sluttdato. I dette tilfellet er produktets leveranse «behandling av sykepengesøknader og utbetaling av sykepenger». Det vil si at målet gjennom hele utviklingen er å behandle sykepengesaker og utbetale sykepenger. Det skal utvikles et forbedret system innenfor eksisterende regelverk. Overordnede mål for teamet som skal undersøkes i dette studiet er for eksempel å « redusere antall saksbehandlere som trengs for å håndtere sykepengesakene i Norge », «komme ut av gammel saksbehandlingsløsning» og å «bidra til en bedre brukeropplevelse for de som er syke i Norge».

1.3 Studiets begrensninger

Studiet er en del av en masteroppgave som varer over 20 uker våren 2022 og tilsvarer 30 studiepoeng; studiet har dermed måttet bli begrenset til å kun se på én case der det utvikles ett produkt. Det er også gjort begrensninger av antall intervjuobjekter og observasjonsperiode. Videre begrensninger er at det kun er observert designerne sine

arbeidsdager, ikke arbeidsdagene til andre roller, og kun to arbeidsdager totalt. Det er kun gitt tilgang til å observere utvalgte kanaler i Slack; dermed er det ikke sikkert observatøren har fått se hele bildet i alle situasjoner.

I tillegg skulle det gjøres observasjoner av en spesifikk brukermedvirkningsaktivitet, men grunnet personvern hensyn kunne ikke denne aktiviteten gjennomføres. Under denne typen observasjon ville det blitt vist personsensitiv data fra søker om sykepenger på dataskjermen til saksbehandler som jeg dermed hadde sett. Søknadsprosessen for å få lov til å gjennomføre observasjon av aktiviteten tok for lang tid i forhold til tidsaspektet på masteroppgaven, og dette ble dermed ikke klarert tidlig nok til å planlegge og gjennomføre observasjon av denne typen aktiviteter. Innholdet har blitt forsøkt dekket gjennom intervjuer istedenfor.

Det finnes lite litteratur spesifikt om produktbasert utvikling. Dermed er det også benyttet litteratur for prosjekter som anses relevant i forhold til arbeidsmetoden til utviklingsteamet.

1.4 Oppgavens oppbygging

Metoden som er benyttet i masteroppgaven er observasjon, intervju, litteraturstudium og analyse. Kapittel 2 gjengir metoden som er benyttet i masteroppgaven. Tidligere studier på relevante temaer er gjengitt i kapittel 3. Videre, i kapittel 4, presenteres casen. Resultatene fra observasjoner og intervjuer om brukermedvirkningsaktiviteter som gjennomføres i utviklingsteamet i NAV presenteres i kapittel 5. Deretter drøftes resultatene for å finne mulige læringspunkter til videre utvikling i kapittel 6. Til slutt presenteres konklusjonen fra studien og forslag til videre arbeid i kapittel 7, etterfulgt av referanser og vedlegg.

Teamet som blir observert lager et produkt der *brukerne* er saksbehandlere av søknader om sykepenger hos NAV. Videre i denne oppgaven vil begrepene *bruker(e)* og *saksbehandler(e)* bli benyttet ut ifra konteksten. I kapittel 3 presenteres *behov-* og *kravspesifikasjoner* som to forskjellige ting. I denne masteroppgaven benyttes likevel *krav* og *behov* om hverandre om ting som brukerne trenger fra systemet.

2 Metode

I dette kapitlet presenteres metodene som benyttes i dette studiet. Til slutt evalueres hver metode med tanke på validitet, reliabilitet og generalisering.

2.1 Forskningsdesign

Et forskningsdesign er «en overordnet plan for studiene som forteller hvordan problemstillingen skal belyses og besvares» (Sander, 2020). Designet for en forskningsoppgave kan enten være kvalitativ eller kvantitativ, avhengig av perspektiv og tilnærming til empiri og teori (Sander, 2020). Når man skal velge forskningsdesign, er det viktig å tenke på ressursene og tiden man har tilgjengelig (Sander, 2020). Det er også vesentlig å velge et design som finner svarene på problemstillingen man ønsker besvart (Sander, 2020).

For denne masteroppgaven er det valgt å gjennomføre et kvalitativt casestudium av ett utviklingsteam i NAV med en induktiv tilnærming. Å studere én case har blitt valgt for å kunne gå mer i dybden i denne ene casen. Studiet gjennomføres gjennom semistrukturerte intervju med teammedlemmer og saksbehandlere, observasjon av skriftlig kommunikasjon i Slack og observasjon av (digitale) arbeidsdager til to teammedlemmer. Det har blitt valgt å observere i kombinasjon med intervjuer for at jeg skal kunne trekke egne slutninger ut ifra observasjoner samtidig som jeg skal få et innblikk i utviklingsteammedlemmene og brukerne sine synspunkter. Når det benyttes flere metoder, kalles det metodetriangulering (Dalen, 2022, pp. 17–22). Det er også benyttet flere datakilder (saksbehandlere og teammedlemmer); dette kalles datatriangulering (Dalen, 2022, pp. 17–22).

Kritiske suksessfaktorer for metodene er som følger:

- Casen er relevant for problemstillingen
- Casen bidrar til svar på problemstillingen
- Funnet litteratur er gjeldende for arbeidsmetoden
- Spørsmålene i intervjuene bidrar til svar på problemstillingen
- Intervjuobjektene svarer ærlig og utfyllende på spørsmålene

I dette studiet hadde jeg ikke noen klar hypotese før jeg gikk i gang med datainnsamling og jeg ønsket å lære underveis i datainnsamlingsprosessen. Dermed benyttet jeg en *induktiv* tilnærming. Ved å benytte en induktiv tilnærming går man fra virkeligheten til teori. Da samler forskeren «inn data med så åpent sinn som overhodet mulig» (Jacobsen, 2021, p. 23). Dette er i motsetning til deduktiv tilnærming der man går fra teori til virkeligheten. Da har forskeren «en klar teori før han eller hun starter datainnsamlingen» (Jacobsen, 2021, p. 23).

Det har blitt benyttet en *kvalitativ* fremgangsmåte fordi tematikken anses som for kompleks til å reduseres til tall og det har vært ønskelig å få frem nyanser. En kvalitativ tilnærming benyttes der virkeligheten anses som «for kompleks til å reduseres til tall, og at man derfor må samle inn informasjon i form av *ord* som åpner for mer nyanserikdom» (Jacobsen, 2021, p. 24). Dette er i motsetning til en kvantitativ tilnærming der man kan måle, og dermed presentere, den sosiale virkeligheten i form av tall (Jacobsen, 2021, p. 24).

Jeg har valgt å benytte en blanding mellom *avstand* og *nærhet*. Konseptet ved å gjennomføre forskningen på avstand vektlegger «at det bør være et skille mellom forskeren og det fenomenet som undersøkes» (Jacobsen, 2021, p. 24). Ved å gjennomføre forskningen i nærhet legges det vekt på at det er et behov for å forstå det andre mener og tror, noe som «åpner for empati og forståelse» (Jacobsen, 2021, p. 24). Gjennom observasjonene er det benyttet avstand i den formen at jeg ikke har interagert med noen under observasjonen. Observasjonsobjektene har likevel visst at jeg har vært til stede, så min tilstedeværelse kan ha påvirket resultatene. Gjennom intervjuene har det blitt benyttet nærhet der jeg har forsøkt å forstå den andre parten sin situasjon.

2.2 Casestudium

Det har blitt gjennomført et casestudium av ett utviklingsteam i NAV; det kan kalles et enkeltcase-studium (Jacobsen, 2021, p. 99). Et casestudium er «en inngående studie av en eller noen få undersøkelsesenheter. [...] I casestudier settes fokuset på én spesiell enhet [...], og de kan avgrenses i rom og tid» (Jacobsen, 2021, p. 97). «Casestudier er relevant når man forsker på organisasjoner der hensikten er å studere systemer, individer, program og hendelser» (Swanson & Holton, 2005, p. 328). Casen som studeres i dette studiet er presentert i kapittel 4.

Under forberedelsene til observasjoner og intervjuer ble jeg bedre kjent med arbeidsmetoden til teamet gjennom samtaler med teammedlemmer; jeg ble kjent med hvilke brukermedvirkningsaktiviteter som gjennomføres av teamet og jeg fant litteratur som utfordret eller støttet denne arbeidsmetoden. Det er viktig å ha forskningsspørsmål som hjelper med å identifisere relevant informasjon om casen slik at man ikke prøver å dekke «alt» (Yin, 2018, p. 29).

Casestudiet har startdato 1. mars 2022 og varighet på seks uker. Fra og med denne datoen ble det gjennomført en endring i utviklingen der antall saksbehandlere som jobber i den nye løsningen ble endret fra ca. 200 til 25 stykk. Det er dermed hensiktsmessig å begynne studiet fra denne datoen. Det er også gjort begrensninger i antall intervjuobjekter og antall ting som skal observeres (arbeidsdager og Slack). Det er viktig å begrense casen (Yin, 2018, p. 31). Dette kan innebære å definere personer (Yin, 2018, p. 31), tid og sted som inkluderes i casen (Jacobsen, 2021, p. 99; Yin, 2018, p. 31). Ved å begrense casen kan det hjelpe med å fastsette omfanget av datainnsamlingen og å separere data som er relevant til studiet fra data som ikke er relevant (Yin, 2018, p. 31).

Det finnes flere bekymringer rundt benyttelse av casestudier (Yin, 2018, pp. 18–22). Noen av disse er at forskeren gjør en for dårlig jobb i å følge systematiske prosedyrer, eller at forskeren tillater at diskutabile funn påvirker retningen til videre funn og konklusjoner som trekkes deretter (Yin, 2018, p. 18).

2.3 Litteraturstudium

Jeg har gjennomført et litteraturstudium for å finne tidligere forskning og materiale som utfordrer eller støtter funnene fra casestudiet. Dette er gjengitt som teori i kapittel 3. Gjennom et litteraturstudium får forskeren et overblikk over hvilken forskning som allerede har blitt gjennomført om temaet. Hvis deler av det planlagte studiet allerede har blitt

gjennomført, kan det spare tid for forskeren og hun kan gå dypere i forskningen sin. Litteraturen har blitt funnet gjennom følgende søkemotorer:

- ISI Web of Science
- Oria
- Scopus
- Google Scholar

Eksempler på søkeord som har blitt brukt er følgende:

- «User involvement» AND «Software development» AND «Agile method» NOT «Health»
- «Software development» AND «Data driven decision support» AND «Agile method»
- «Software development» AND «User testing» AND «Agile method»

Noen kombinasjoner av søkeord har gitt for mange resultater til å kunne gå gjennom alle. I slike tilfeller har jeg enten lagt inn flere, mer spissede, søkeord, eller valgt artiklene ut ifra dato, utgiver eller forfatter, der de nyeste artiklene har blitt vurdert til å være mest relevant. Litteratur har også blitt samlet fra veileder til masteroppgaven, veiledere i NAV, forelesere på NTNU og gjennom eget nettverk. Litteratur som er eldre enn fra år 2010 har blitt ansett som mindre relevant til denne masteroppgaven da utviklingsprosesser for IT-systemer utvikler seg raskt. Likevel har Iivari og Molin-Juustila (2009) benyttet litteratur fra så tidlig som 1991 som påpeker relevante problemer rundt brukermedvirkning i IT-utvikling.

2.4 Observasjon

I dette studiet er det gjennomført åpen, ikke-deltakende observasjon. «Observasjon dreier seg i bunn og grunn om å registrere personers og gruppers *atferd*» (Jacobsen, 2021, p. 165). Observator observerer det som faktisk blir gjort «i stedet for å spørre mennesker om hva de gjorde» (Jacobsen, 2021, p. 165). At observasjonen er åpen vil si at de som observeres vet at de blir observert (Jacobsen, 2021, p. 166). Dette kan påvirke deres atferd, noe som videre kan påvirke resultatet fra observasjonen (Jacobsen, 2021, p. 166). Ikke-deltakende observasjon vil si at observator holder «avstand til dem han eller hun observerer» (Jacobsen, 2021, p. 167) og interagerer dermed ikke med de som observeres. Ved å observere unngår man at mennesker ikke husker hva de gjorde, eller at de snakker usant (Jacobsen, 2021, p. 165). Observator kan stole på at det som observeres faktisk skjer (Olseng & Sundbye, 2021), og man kan tolke situasjonene underveis.

Jeg lagde et observasjonsskjema, slik som vist i vedlegg A. Innholdet i skjemaet er valgt gjennom funn fra litteraturen og gjennom samtaler med to utviklingsteammedlemmer som jobber aktivt med brukermedvirkning. Skjemaet var åpent for endringer underveis i studiet hvis det skulle vise seg at annen problematikk var mer interessant å fordype seg i. Samme observasjonsskjema ble benyttet til alle observasjonskanalene, men de fikk tildelt hver sin fil. Dette skulle gjøre det lettere å holde datakildene fra hverandre i analyseprosessen.

Observasjonen ble gjennomført ved å observere kommunikasjonen mellom utviklingsteamet og saksbehandlerne over Slack og ved å skygge to designere gjennom hver sin arbeidsdag. Jeg har fått tilgang på noen kanaler i Slack for å observere kommunikasjonen der. Hvilke kanaler jeg har fått tilgang på har blitt valgt av designerne.

Jeg skygget de to designerne én arbeidsdag hver. Jeg observerte alle møter og diskusjoner som foregikk disse dagene. Mitt oppmøte måtte være godkjent av alle deltakerne i de digitale rommene før jeg kunne delta. Målet med disse skyggedagene var å se hvordan tiden til utviklingsteamet ble brukt gjennom dagen og for å se hvordan brukerne av systemet blir hensyntatt i avgjørelser. Saksbehandlere ble ikke observert gjennom noen aktiviteter.

Underveis i observasjonsperioden fylte jeg inn observasjoner i observasjonsskjemaet. Etter observasjonsperioden strukturerte jeg dataene og analyserte dem slik som forklart i kapittel 2.6.

2.5 Intervjuer

Det har blitt gjennomført semistrukturerte intervjuer med seks utviklingsteammedlemmer og fem saksbehandlere som er brukere av systemet og som kun jobber i dette systemet. Semistrukturerte intervju inneholder forhåndsbestemte spørsmål, der alle intervjuobjekter blir stilt de samme spørsmålene (AcademicWork, 2021). Intervjueren kan stille oppfølgingsspørsmål basert på svarene til intervjuobjektene (AcademicWork, 2021). Semistrukturerte intervjuer ble valgt som intervjuform for å ha muligheten til å stille oppfølgings- og oppklaringsspørsmål hvis jeg skulle se at det var nødvendig.

Jeg ønsket å intervjuer minst fem fra «hver side» da det ville gi et greit innblikk i hvordan brukermedvirkning foregår i utviklingsprosessen, sett fra begge sider; jeg ønsket at intervjuobjektene har vært aktive i brukermedvirkningsprosessen i varierende grad. Dette var for at funnene skulle være mer generaliserbare, se kapittel 2.7.4.

Formelle stillinger til deltakerne fra utviklingsteamet er designer, jurist, utvikler og produkteier. Blant saksbehandlere som ble intervjuet var det både «vanlige» saksbehandlere og såkalte «supersaksbehandlere».

Det ble holdt to forskjellige typer intervjuer. Det ene type intervju var et generelt intervju om brukermedvirkningsaktivitetene som gjennomføres av utviklingsteamet og ble holdt individuelt med to teammedlemmer som har mye med brukermedvirkning å gjøre. Spørsmålene som ble stilt er vist i vedlegg B og intervjuene varte mellom 40 og 60 minutter. Det andre type intervju hadde som hensikt å bli kjent med én spesifikk brukermedvirkningsaktivitet der deltakere fra utviklingsteamet observerer saksbehandlere i arbeid i den nye løsningen. Dette intervjuet ble holdt med fem saksbehandlere som kun jobber i ny løsning og fire utviklingsteammedlemmer. Spørsmålene som ble stilt er vist i vedlegg C og intervjuene varte mellom 15 og 25 minutter. Alle intervjuene ble holdt, og tatt opp, via Zoom. Opptakene ble transkribert og de transkriberte intervjuene ble sendt til intervjuobjektene for godkjenning der uklarheter ble oppklart.

Hver person sitt bidrag har blitt anonymisert fordi det vurderes at det ikke gir mer verdi å vite hvem som har sagt hva. Det er også vurdert at oppgivelse av stilling enkelt kan spores tilbake til enkeltpersoner i og med at teamet har relativt få i hver stilling.

2.6 Analyse

All analyse av datamaterialet som er samlet inn gjennom observasjon og intervjuer er gjennomført ved bruk av induktiv koding gjennom programmet NVivo. Induktiv koding er å

lage «koder basert på den kvalitative dataen» (Medelyan, 2019). Å «kode» data er å gi «merkelapper» og organisere den kvalitative dataen for å identifisere forskjellige temaer og forhold mellom dem (Medelyan, 2019). De transkriberte intervjuene og dokumentene fra observasjonene har blitt lastet ned til NVivo. Deretter leste jeg gjennom hvert intervju, lagde koder og sorterte fortløpende. I mitt tilfelle har kodene vært basert på spørsmålene i intervjuene eller mer spisset inn mot funn og elementer som har blitt sagt av intervjuobjektene. Koder fra observasjonsdagene og fra observasjonen av kommunikasjon i Slack er gjengitt i Tabell 2.1. Kodene fra de generelle og de spissede intervjuene er gjengitt i Tabell 2.2.

Funnene ble videre analysert og gruppert ut ifra kodene. Funn som tidligere lå under samme kode kunne bli splittet opp og satt sammen med andre funn ut ifra sammenheng og interessante vinklinger; dette utgjør resultatkapittelet, kapittel 5.

Koder fra observasjon	
Observasjonsdagene	Observasjon av kommunikasjon i Slack
Agenda	Blir brukerne hørt i sakene de tar opp
Brukerne hensyntatt av utviklingsteamet	Kommunikasjon fra utviklingsteam til brukerne
Brukerne hensyntatt i løsningen	Kommunisere problemer
Kommunikasjon gjennom dagen	Lansering av nye løsninger
Kommunisere problemer	Oppklaring av saker fra utvikler sin side
Menneskelig atferd	Responstid fra utviklingsteamet
Språkbruk	Skjønner utviklingsteamet brukernes behov
Tidsbruk i løpet av dagen	Språkbruk

Tabell 2.1: Koder fra observasjon

Koder fra intervjuer	
Generelle intervjuer	Spissede intervjuer
Alternative brukermedvirkningsaktiviteter	Agenda for treffene
Arbeidsflyt	Andre nyttige brukermedvirkningsaktiviteter
Autonomitet	Antall treff de har vært med på
Bedre løsninger på problemer	Er treffene nyttige
Brukerne hensyntatt av utviklingsteamet	Gjennomføring
Definisjon brukermedvirkning	Oppdaget feil i systemet
Definisjon smidig metode	Problemer med den andre parten under treffene
Holdninger til brukermedvirkning	Tydlig kommunikasjon fra saksbehandlerne
Hva kunne blitt gjort for at problemene ikke oppstår med det første	Tydlig kommunikasjon fra utviklingsteamet
Informasjonslagring	
Inntrykk	
Hva gir minst verdibidrag til brukerne	
Motivasjon til brukermedvirkning	
Problemløsninger	

Relasjonsbygging	
Representasjon av brukere i brukergruppa	
Saksbehandlers inntrykk	
Slack	
Hva gir størst verdibidrag til brukerne	
Utfordringer	
Utviklingsteamets arbeidsdag	

Tabell 2.2: Koder fra intervjuene

2.7 Evaluering av forskningsdesign

Dette delkapitlet går inn på validitet, reliabilitet og generalisering ved metodene som har blitt benyttet. For at forskningen skal kunne gi grunnlag for beslutninger er det viktig at den har høy validitet og reliabilitet, der validitet er det viktigste (Sander, 2019). Gjennom validitet måler man om undersøkelsen «faktisk gir svar på det eller de spørsmålene vi har stilt» (Jacobsen, 2021, p. 17). Validiteten kan deles inn i intern og ekstern validitet. Den interne validiteten omhandler om egen data dekker de konklusjonene som trekkes (Jacobsen, 2021, p. 17). Ekstern validitet omhandler om resultatene fra ett case også er gyldige i andre sammenhenger (Jacobsen, 2021, p. 17). Gjennom reliabilitet måler man om undersøkelsen er til å stole på, og om man ville fått tilnærmet samme resultat dersom man gjennomfører forsøket to ganger (Jacobsen, 2021, p. 17). Generalisering vil si «å påstå at funn basert på studier av noen få, også gjelder for alle» (Jacobsen, 2021, p. 86). For å diskutere generalisering, må det sees på populasjonen og utvalget. En populasjon er «alle de undersøkelsesenheterne vi ønsker å uttale oss om» (Jacobsen, 2021, p. 87). «Et utvalg kan defineres som en undergruppe av populasjonen» (Jacobsen, 2021, p. 87). For å øke muligheten for generalisering er det viktig å få på plass et representativt utvalg og at utvalget er av en viss størrelse. Utvelgelsen skal også skje ved å velge tilfeldige representanter for å være generaliserbar.

2.7.1 Evaluering av casestudium

Validitet

Validiteten til en metode går som sagt på om undersøkelsen faktisk gir svar på det man lurer på. I dette casestudiet har det blitt benyttet en induktiv tilnærming. Dermed har forskningsspørsmålene blitt utformet for at casen skal kunne besvare disse spørsmålene. Validiteten til casestudiet kan dermed vurderes til å være høy. Videre har det blitt forøkt å øke validiteten gjennom å ha flere datakilder (intervju og observasjon, der intervjuene har blitt gjennomført med begge parter i brukermedvirkningsprosessen). Å ha flere datakilder anbefales også av Yin (2018, p. 44) for å øke validiteten til et casestudium.

Reliabilitet

Reliabiliteten til et casestudium kan avhenge av hvor godt gjennomføringen beskrives (Yin, 2018, p. 46). Det er dermed viktig å dokumentere godt hvordan studiet gjennomføres slik at andre forskere skal kunne gjennomføre et likt studium i senere tid. Teamet som studeres i dette casestudiet endrer arbeidsmetode hele tiden, så arbeidsmetoden de benyttet da dette

studiet ble gjennomført, kan være utdatert noen måneder etterpå. Resultatene er dermed ikke etterprøvbare ved å gjennomføre studiet en gang til i senere tid. Likevel kan studiet oppnå høy reliabilitet ved at fremgangsmåter og gjennomføringer beskrives godt nok (Yin, 2018, p. 46).

Generalitet

Det er kun tatt for seg ett case. «Det er vanskelig å generalisere statistisk fra en case til andre case» (Jacobsen, 2021, p. 100). Hvis IT-utviklingsteam i Norge er populasjonen, så er denne ene casen et veldig lite utvalg. Dette påvirker muligheten for generalisering i negativ forstand. Hvis populasjonen er «alle utviklingsteam i NAV», er den gjeldende casen et større utvalg, men fortsatt så lite at det er lav mulighet for generalisering av resultatene.

2.7.2 Evaluering av litteraturstudium

Validitet

Validiteten til et litteratursøk kan avhenge av søkeordene som benyttes. Ved å benytte seg av feil eller misledende søkeord, kan man ende opp med ubrukelig eller bias litteratur. Hvilke forskningsdatabaser som benyttes er også vesentlig med tanke på validitet av litteraturen. Hvis forskningsdatabasen tillater «hvem som helst» å publisere i databasen, kan informasjonen være feil. Dette har vært i fokus gjennom litteraturstudiet som har blitt gjennomført til denne masteroppgaven, og validiteten vurderes dermed til høy.

Reliabilitet

Reliabilitet kan påvirkes av hva forskningsdatabasene tillater at publiseres. Videre avhenger reliabiliteten av hvor godt forfatterne av litteraturen beskriver betingelser og begrensninger for sin egen forskning. Det avhenger også av om forfatterne, forlagene og/eller forskningsdatabasene er til å stole på. Det er dermed viktig å beskrive situasjonene fra litteraturen når det benyttes inn i et nytt studium, slik som inn i denne masteroppgaven, slik at leseren blir informert om tilfellene der litteraturen er gjeldende.

Generalisering

Muligheten for generalisering av funn gjennom litteratursøk må vurderes opp mot hvert enkelt forskningstilfelle. Forskeren må vurdere om hvert funn fra litteraturen kan anvendes i egen forskning. Hvis situasjoner i litteraturen beskrives i tilstrekkelig grad, er det enklere å vurdere om funnene kan generaliseres til egen forskning. Derimot hvis det ikke beskrives godt nok, vet ikke mottakeren om funnene fra litteraturen er gjeldende for ens egen forskning.

2.7.3 Evaluering av observasjon

Validitet

I dette studiet er det benyttet åpen observasjon. Validiteten til observasjon kan dermed påvirkes av om personene som observeres oppfører seg på en annen måte fordi de vet at de blir observert. Dette var tilfellet både under observasjon av arbeidsdagene, og under observasjon av kommunikasjon i Slack. Startdatoen for observasjon i Slack var *etter* at alle i

kanalene var spurt om det var greit å observere den gjeldende kanalen. Dette påvirker validiteten i negativ forstand.

Validiteten kan også påvirkes av hvordan observasjonen blir gjennomført. Hvis det for eksempel kun er én observatør og veldig mange inntrykk som én person ikke klarer å observere på egenhånd, kan det bli vanskelig å holde høy validitet fordi inntrykk kan bli oversett. I observasjonen av arbeidsdagene var det oversiktlig for én observatør å observere. Observasjonen av den skriftlige kommunikasjon i Slack var også oversiktlig da det ble gjennomført i intervaller som observatør bestemte selv. Dermed har observasjonen som har blitt gjennomført i dette studiet blitt vurdert til å ha medium til høy validitet.

Reliabilitet

Reliabiliteten ved observasjon kan påvirkes ved at observasjonsobjektene kan oppføre seg på en litt annen måte når man vet at man blir observert (Olseng & Sundbye, 2021). Hvis de gjennomfører oppgavene på en annen måte enn til vanlig fordi de blir observert, vil det ikke gjenspeile den virkelige situasjonen. Hvilke kommunikasjonskanaler jeg har fått innsyn i kan påvirke hvor virkelighetsnært observasjonen er. Likevel kan observasjon ha høy reliabilitet fordi observatør kan legge merke til elementer som ikke oppdages av de som er midt oppi det. Observasjonen kan også ha høy reliabilitet ved at observatør selv kan tolke og tyde situasjonene som oppstår og koble hendelsene til det som ønskes målt.

Generalisering

Det er kun gjennomført observasjon av to arbeidsdager der det er skygget to forskjellige personer. Dette kan anses som et lite utvalg, gitt at populasjonen er alle arbeidsdager i utviklingsprosessen. I samarbeid med to teammedlemmer er det forsøkt å finne dager som kan observeres der det er noe brukermedvirkning på dagsagendaen, og dagene er dermed ikke trukket tilfeldig. Det er uvisst om disse dagene er representative for «hverdagen». Alle disse aspektene påvirker mulighet for generalisering av skyggedagene til å være lav.

Når det kommer til observasjonen i Slack, så har kanalene jeg har fått innsyn i blitt valgt av teammedlemmer. Dermed er heller ikke dette tilfeldig valgt. Det er varierende grad av aktivitet i hver kanal og det forekommer kommunikasjon på andre plattformer, så populasjonen kan ikke defineres av *antall* kanaler. Det er dermed uvisst hvor stort utvalget er i forhold til populasjonen, men det anses som et lite utvalg. Kanalene jeg har fått innsyn i er en del av én kommunikasjonskanal i forhold til alle kommunikasjonsmåter mellom brukere og utviklingsteammedlemmer (mail, ansikt til ansikt, telefonsamtale, videosamtale, ikke-verbal kommunikasjon, osv.). Kanalene i Slack anses som et lite til middels representativt utvalg. Likevel vurderes generaliserbarheten av kommunikasjonen i Slack til å være relativt lav. Den totale generaliserbarheten av observasjonene blir dermed vurdert til å være lav.

2.7.4 Evaluering av intervjuer

Validitet

Validiteten til intervjuene kan avhenge av måten spørsmålene blir stilt. Hvis spørsmålene er ledende, kan man få misledende svar i forhold til hva som faktisk er hensikten med spørsmålene. Hvis det stilles for åpne spørsmål, kan man få svar på andre ting enn det man

er ute etter. Dette kan rettes opp ved å stille mer konkrete oppfølgingsspørsmål og å være tydelig i spørsmålsstillingen. Validiteten avhenger også av kunnskapen til intervjuobjektene på temaet intervjuet omhandler. Alle disse aspektene har vært med i utvelgelsen av intervjuobjekter og i vurderingen av spørsmålene som skulle stilles under intervjuene; både forhåndsbestemte spørsmål og oppfølgingsspørsmål. Dermed vurderes validiteten av intervjuene til å være middels til høy.

Reliabilitet

Reliabiliteten til intervjuene kan avhenge av om man intervjuer folk fra mange nok sider av temaet som skal studeres. Dette har blitt tatt hensyn til i dette studiet gjennom å intervjuer folk fra flere sider av utviklingen. Det kunne ha blitt gjennomført intervjuer med flere folk fra hver side, så reliabiliteten fra dette aspektet vurderes til å være middels.

Hvis det stilles ledende spørsmål kan man få svar som peker i én spesifikk retning, noe som kan føre til at man får villedende svar. Spørsmålene bør stilles på en entydig måte slik at både intervjuer og intervjuobjektet er enige om hva det faktisk spørres om. Dermed er det større sannsynlighet at intervjuobjektet svarer på det som det spørres om. Det er lagt vekt på dette under intervjuene til dette studiet, så reliabiliteten fra dette aspektet vurderes til å være høyt.

Erfaringene til intervjuobjektene og tolkningen til intervjuer er også vesentlig for måling av reliabiliteten. For å oppnå høy reliabilitet må intervjuobjektene dele så mye som mulig for at intervjuer skal få med seg hele bildet og dermed kan vurdere hva som er relevant informasjon. Totalt for intervjuene vurderes reliabiliteten til å være medium til høy.

Generalisering

Intervjuobjektene kommer både fra teamet og fra saksbehandlerne, og jeg ønsket at de skulle være aktive i brukermedvirkningsprosesser i varierende grad. Dette var for at funnene skulle være mer generaliserbare enn hvis man for eksempel kun intervjuet folk fra den ene siden og at alle disse var veldig aktive når det kommer til brukermedvirkning. Generalisering på dette punktet kan anses som middels.

Hvis man ser på deltakere i utviklingen av systemet som populasjonen så består utvalget av 11 av totalt 50 deltakere, noe som gir 22% dekning. Dette kan anses som en grei dekningsgrad for å kunne generalisere resultatene i forhold til utviklingsdeltakerne. Men, intervjuobjektene har blitt valgt ut ved at de selv meldte interesse for å bli intervjuet; dette går imot et av kravene for generalisering som sier at utvalget skal være tilfeldig valgt. De som har meldt sin interesse kan være godt over middels interessert i å bidra, noe som ikke trenger å være en holdning for alle deltakerne i utviklingen. Dermed vurderes muligheten for generalisering innad i utviklingen som middels.

3 Teori

Dette kapitlet presenterer teori funnet ved litteratursøk og som anses relevant for brukermedvirkning i NAV. Først forklares brukermedvirkning og forskjellige aspekter innenfor dette temaet. Videre introduseres forskjellige arbeidsmetoder, herunder fossefallsmetoden og smidig metode. Til slutt introduseres noen team- og organisasjonsmodeller som anses relevante.

Deler av litteraturen som er benyttet til denne masteroppgaven kan være ment spesifikt mot prosjekter, noe som avviker fra arbeidsmetoden til case-teamet. Tilfeller der det er usikkert om teorien kan anvendes på annet enn prosjekter er det presisert ved å omtale utviklingen som et prosjekt.

3.1 Brukermedvirkning

3.1.1 Definisjoner og viktigheten av brukermedvirkning

I denne oppgaven er brukermedvirkning å inkludere sluttbrukere av et system eller en service i utvikling eller forbedring av systemet eller servicen for å lage et bedre produkt. Ved å involvere brukere får man mer kunnskap om problemet som skal løses (Civil Service India, 2021). Brukermedvirkning kan benyttes på flere områder. I utviklingsprosjekter kan det benyttes når man skal finne mål for en løsning, hvilken prioritet hver utviklingsaktivitet bør ha, eller når man skal bestemme hvordan forskjellige elementer i en løsning skal være, både høyt og lavt prioritert. Brukerne bør få god informasjon om servicen eller produktet som skal utvikles for å kunne medvirke på en god måte.

Robertson og Robertson (2013, p. xxi) legger vekt på hvor viktig det er å være sikker på at man løser det riktige problemet og at man må gjøre en bedre jobb med utarbeidelse av kravspesifikasjoner. Ved å involvere brukerne i utviklingen kan utviklingsteamet unngå å gjøre feil oppgaver (Liu & Wang, 2014, p. 1151). «Forskjellige interessenter kan ha forskjellig syn på hva som gir verdi» (Johansen et al., 2019, p. 7). Ved å involvere brukerne, er det lettere å samkjøre målene til sluttbruker og prosjektleder (Liu & Wang, 2014, p. 1151). *Måten* brukerne blir involvert på er likevel viktig (Andersen, 2018, p. 410). Det er et høyere fokus på å respondere kjapt enn å dokumentere godt når man jobber smidig (Saeeda et al., 2020, p. 2). Dette kan føre til at nødvendige krav fra brukeren kan gå ubemerket (Saeeda et al., 2020, p. 2).

Det viktigste ved softwareutvikling er verdien det gir til eierne eller brukerne (Robertson & Robertson, 2013, p. xxi). Hvis utviklingen av et produkt bruker for lang tid og mer penger enn det som er bestemt i budsjettet, men systemet gir stor verdi til eierne er det mye bedre enn det mulig billigere alternativet (Robertson & Robertson, 2013, p. xxii).

3.1.2 Når bør brukerne involveres?

Forskere advarer mot risikoen ved å inkludere brukere i innovasjonsprosessen fordi det medfører en risiko å være avhengig av kundene sine synspunkter og at det kan oppstå misforståelser mellom utviklere og kundene (Bosch-Sijtsema & Bosch, 2015, p. 793), samt andre ting. Johansen et al. (2019, pp. 56–57) sier at det er «usikkerhet rundt fremtidige

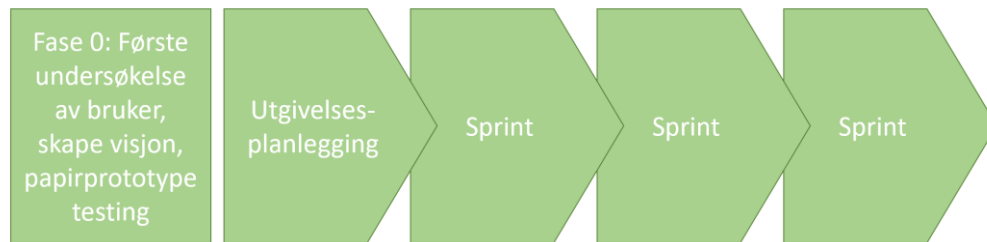
behov» i begynnelsen av et prosjekt med tanke på mål, leveranser og «fremtidige krav fra kunder og interessenter». Dette kan tyde på at å involvere brukerne tidlig i utviklingsprosessen kan være gunstig for å finne ut av dette.

Når man benytter smidige metoder er det anbefalt å holde brukerne kontinuerlig involvert (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32). På denne måten kan brukerne gi tilbakemeldinger gjennom hele utviklingstiden og leveransen kan bli bedre enn hvis utviklingsteamet glettet hva brukerne ønsket. Men, brukere har ikke alltid tid til å delta i brukermedvirkningsaktiviteter (Beyer, 2010, p. 18). Det er likevel viktig å huske at kontinuerlig involvering av brukere som ikke har så mye kunnskap om hvilke kravspesifikasjoner som kan funke og ikke, går i konflikt med softwareeksperter og kan føre til fiasko når det kommer til tekniske aspekter ved produktet (Saeeda et al., 2020, p. 2).

Ifølge Schmitz et al. (2019, p. 32) er softwareprosjektteam anbefalt å inkludere brukere i spesifikke aktiviteter, slik som kravspesifikasjoner, testing og opplæring. Ifølge Abelein og Paech (2014, p. 95) er brukermidvirkning i softwareutvikling ansett som essensielt for at softwaresystemet skal bli en suksess. Robertson og Robertson (2013, p. xxii) sier at siden software er en løsning så må man forstå det virkelige problemet før man kan løse det. Det kritiseres at brukere blir kontaktet først etter basisfunksjonalitet er definert, hvis de blir kontaktet i det hele tatt (Iivari & Molin-Juustila, 2009, p. 55). Likevel er det en vid aksept for at brukere burde bli involvert i IT-utvikling (Iivari & Molin-Juustila, 2009, p. 55). Brukere og utviklere bør anses som likeverdige parter der begge bidrar med sin spesifikke ekspertise (Iivari & Molin-Juustila, 2009, p. 55).

3.1.3 Brukermidvirkningsmodell

Beyer (2010, pp. 27–48) presenterer en struktur for brukersentrert smidig utvikling. Denne strukturen er gjengitt i Figur 3.1. I fase 0 skal teamet kartlegge nåværende situasjon og bli kjent med brukernes behov som videre hjelper dem med å forstå hvordan løsningen bør være. Første steg innenfor denne fasen er «å intervju brukerne på arbeidsplassen deres, se på dem i arbeid og snakke med dem om hva de gjør og hvorfor» (Beyer, 2010, p. 28). Det nåværende fokuset er på «hvordan folk gjennomfører arbeidet, hva de prøver å oppnå, hvordan de går frem og hva som stopper dem med å få gjort jobben» (Beyer, 2010, p. 28). Alle teammedlemmer kan trenes opp til å være med på kartleggingen, noe som sørger for at alle på teamet forstår brukernes problemer på et følelsesmessig plan. Videre, under utgivelsesplanlegging, skriver man brukerhistorier, estimerer kostnader og planlegger utgivelser. Brukerhistoriene bør inneholde høy-nivå beskrivelser av funksjonalitet eller oppførsel (Beyer, 2010, p. 41) som vil si at det er enkle beskrivelser uten for mye detaljer. Underveis som brukerhistoriene skrives, estimerer teamet kostnadene for å implementere dem (Beyer, 2010, p. 42). Deretter organiseres brukerhistoriene i sprinter (Beyer, 2010, p. 43), se kapittel 3.2.5.



Figur 3.1: Grunnleggende struktur for brukersentrert smidig utvikling (Beyer, 2010, p. 27)

«Folk lager artefakter som skal hjelpe dem med å gjennomføre arbeidet sitt. [...] Hvert artefakt gir innblikk i hvordan brukeren går frem for å gjøre arbeidet og hva de trenger for å støtte det.» (Beyer, 2010, p. 32). Et artefakt kan for eksempel være en kontaktliste på papir, eller at man gjennomfører utregninger ved hjelp av en kalkulator istedenfor å benytte systemet (Beyer, 2010, p. 32). Artefaktene kan føre til at arbeidet blir gjennomført forskjellig fra hvordan systemet ville gjort det, og dermed også forskjellig fra hvordan det er ønsket. Det kan dermed være nyttig å fange opp disse artefaktene (Beyer, 2010, p. 32) slik at man kan implementere dem i løsningen eller utnytte at de benyttes som eksterne hjelpemidler.

3.1.4 Utvelgelse av brukere til medvirkningsprosessen

Rasmussen et al. (2011, p. 176) fant at brukere ofte blir valgt til å medvirke basert på om de kan bidra i fremdriften av IT-utviklingen heller enn om de er representative for brukergruppa av systemet. Hvis dette er tilfellet kan produktet ende med å kun oppfylle behovene til *del*er av brukergruppa (Rasmussen et al., 2011, p. 176). Videre sier de at det er viktig at de viktigste delene av brukergruppa bør være representert (Rasmussen et al., 2011, p. 176). De utforsker forskjellige kriterier som har blitt benyttet for å velge brukere til samarbeid i IT-utvikling ved å intervjuer folk fra fire IT-organisasjoner og to av deres kundeorganisasjoner. De fant at brukere blir valgt til å bidra i utviklingen basert på brukernes profesjonelle kunnskap, deres personlige egenskaper og den totale sammensettingen av brukere (Rasmussen et al., 2011, p. 186). For lengre samarbeidsperioder med brukere veier det tyngst at brukerne kan bidra aktivt og gi direkte fremgang i utviklingen (Rasmussen et al., 2011, p. 186).

3.1.5 Behov- og kravspesifikasjoner

Kravbehandling (Requirement Engineering) er kritisk, nettopp fordi krav, eller behov, spiller en viktig rolle i alle softwareutviklingslivssykluser (Rasheed et al., 2021, p. 1). Beck og Andres (2005, p. 44) påpeker at «krav» defineres som «noe pålagt eller obligatorisk» og de mener at det er feil å bruke dette ordet i IT-utvikling. Videre sier de at hvis man har tusen sider med «krav», og realiserer de riktige 5-20% så vil man mest sannsynlig oppnå alle businessmålene som er tenkt for systemet (Beck & Andres, 2005, p. 44). I denne masteroppgaven benyttes likevel *krav* og *behov* om hverandre om ting som brukerne trenger fra systemet.

Beyer (2010, p. 18) hevder at hvis brukere blir spurt om deres arbeidsoppgaver og om hva brukerne trenger, vil de ikke klare å gjengi dette detaljert. Videre hevder han at kravspesifikasjonene som gis fra brukerne på denne måten er uspesifikke og uferdige. Dette kan underbygge at observasjon av brukere og hvordan de jobber kan være en lurere

fremgangsmåte enn å *kun* stille dem spørsmål. Kanskje er spørsmålstilling en god start, men at man videre bør benytte andre metoder.

Beyer (2010, p. 18) hevder at brukere ønsker å være hjelpsomme, noe han mener kan medføre at de responderer på småting i systemet som utviklingsteamet ønsker respons på istedenfor å få fram at de egentlig trenger et annet system, hvis det skulle være tilfellet. Dette kan underbygge at direkte brukertesting (se kapittel 3.1.6) kanskje ikke helt er veien å gå fordi utviklingsteamet kun får svar på de tingene de direkte lurer på. Det kan hende at det de lurer på ikke har noe relevans for hva som faktisk trengs.

Ifølge Redström (2005, p. 123) bør man være forsiktig med å overdrive tilpasning av systemet med kunnskap fra brukerne. Hvis man jobber for hardt med å lage en perfekt overenstemmelse kan det føre til at utviklerne ikke har nok rom til å improvisere (Redström, 2005, p. 129).

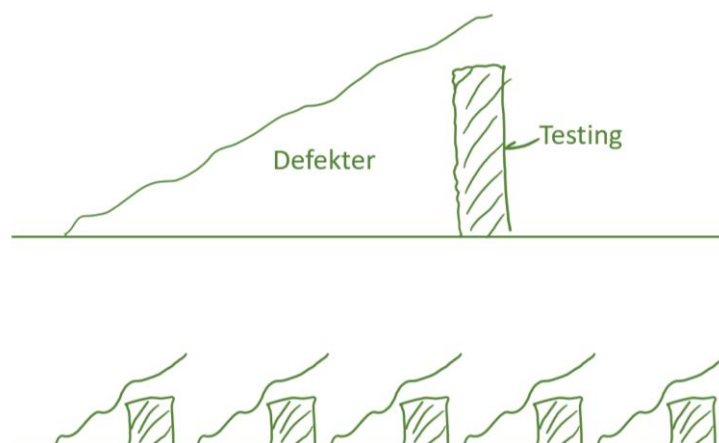
3.1.6 Testing og innhenting av data om brukeropplevelse

Brukergrensesnittet og brukeropplevelsen (user experience, UX) kan testes for å finne ut av om den er som den burde være, og om systemet har den funksjonaliteten som trengs. Dette kan gjennomføres på forskjellige måter, for eksempel gjennom brukertester. Én måte er at man presenterer spesifikke aspekter ved en løsning til brukere av systemet for å bekrefte eller avkrefte om løsningen fungerer eller er god nok¹. Dette er gjerne direkte rettet mot en spesifikk funksjonalitet, og er konstruerte tester. Det vil si at testen vil sjekke om en spesifikk funksjonalitet fungerer som tiltenkt eller ikke, men den vil ikke sjekke om denne funksjonaliteten faktisk trengs. En annen måte å gjennomføre brukertesting, er gjennom å observere hvordan kontinuerlige leveranser blir mottatt og ha kontinuerlig dialog med brukerne av systemet og dermed teste funksjonaliteten i et virkelig miljø.

Softwarekvalitet er viktig i utvikling av softwaresystemer (Rosenberg et al., 2020, p. 109). Det vil si at systemet passer til sin hensikt. Dette kan utforskes gjennom softwaretester. Da tester man systemet, eller spesifikk funksjonalitet, basert på individuelle krav (Rosenberg et al., 2020, p. 109). Softwaretesting, der man tester koden til funksjonaliteter, er fundamentalt for enhver softwareutvikling og er i dag en nødvendig del av utviklingsprosessen (Rosenberg et al., 2020, pp. 108–109). Utviklingsteamet kan ha dedikerte testere som koder automatiske aksepttestscripts (Rosenberg et al., 2020, p. 109). I den sammenheng konstrueres tester med akseptkriterier og funksjonalitetens design må suksessfullt komme seg gjennom en slik test (Tatale & Chandra Prakash, 2020, p. 268). Ideelt sett jobber disse testerne tett med utviklerne. Veldefinerte testprosedyrer er spesielt viktig i hurtig utvikling med store team (Rosenberg et al., 2020, p. 109). Å produsere feilfri kode anses som så å si umulig (Beck & Andres, 2005, p. 97). Ved å gjennomføre sosiale øvelser i Extreme Programming (se kapittel 3.3.2), som for eksempel parprogrammering, kan man redusere sannsynligheten for å produsere feil (Beck & Andres, 2005, p. 98). Ved å teste hyppig kan man redusere kostnader

¹ (*What Is Usability Testing?*, 2022)

og defekter. Dette er illustrert gjennom Beck og Andres' (2005, p. 99) figur som er gjengitt i Figur 3.2.



Figur 3.2: Mengden defekter som oppstår ved sjelden testing (øverst) versus hyppig testing (nederst) (Beck & Andres, 2005, p. 99)

Datadreven beslutningsstøtte benyttes til blant annet strategisk forretningsinnsikt (Business Intelligence), statisk og sanntids prestasjonsmåling og ledelse av kundeforhold (Power, 2008, p. 149). Kvantitativ data kan samles inn for eksempel gjennom spørreundersøkelser og monitorering av hvordan produktet brukes av brukerne; men man må nøye seg «med å måle ganske enkle forhold» (Jacobsen, 2021, p. 135) så det gir en begrenset forståelse av situasjonen. Nøkkelen for at datadrevet beslutningsstøtte skal bli en suksess er å ha lett og rask tilgang til en stor mengde med nøyaktig, velorganisert, flerdimensjonal data (Power, 2008, p. 150). Beslutninger som tas med bakgrunn i data benyttes gjerne for å ha bedre grunnlag for beslutningene.

Universell utforming handler om å tilrettelegge slik at alle har mulighet til å fungere på egenhånd i hverdagen². Dette kan være gjennom å lage tekniske løsninger som alle, selv om man har redusert syn, hørsel eller motorikk, har forutsetninger for å kunne bruke. Web Content Accessibility Guidelines (WCAG) er en standard som setter krav til universell utforming for nettsteder og apper og «et nettsted skal følge kravene om universell utforming»³. Kravene gjennom WCAG omfatter alternativt format, kontraster, navigasjon, presentasjon, skjemautfylling, struktur, styring av lyd og tastaturbetjening⁴. Hvilke av kravene i WCAG «som gjelder i praksis, varierer etter appens funksjonalitet og kompleksitet»⁵, og virksomheter «bør ta stilling til og dokumentere hvilke krav eller deler av krav i standarden som ikke er aktuelle»⁵.

² (Intro til universell utforming, 2022)

³ (Nettsteder | Tilsynet for universell utforming av ikt, 2022)

⁴ (WCAG sortert etter tema | Tilsynet for universell utforming av ikt, 2022)

⁵ (Universell utforming av apper | Tilsynet for universell utforming av ikt, 2022)

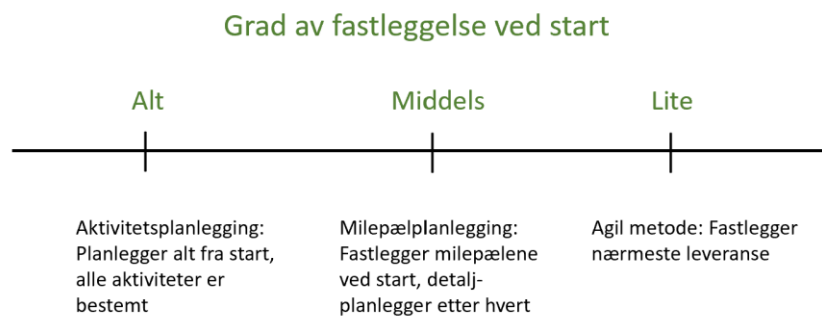
3.2 Arbeidsmetoder

3.2.1 Key performance indicators (KPI)

«Key performance indicators (KPI) er kvantifiserbare og strategiske målinger som reflekterer virksomhetens kritiske suksessfaktorer» (Zhu et al., 2018, p. 82). KPI-er kan benyttes til å måle yteevnen til ansatte, for eksempel gjennom antall arbeidsoppgaver som gjennomføres i løpet av en gitt periode. Hensikten med å rapportere prestasjoner (performance) er å vise hvilke aktiviteter eller handlinger som mottakeren av rapporteringen må overveie (Kerzner, 2017, p. ix). Ved å måle feil ting, kan det gjøre det vanskelig å skjønne hvilke beslutninger som må tas (Kerzner, 2017, p. ix). Dermed kan feil valg av KPI-er som skal måles føre til at ansattes atferd motvirker den atferden som er ønsket gjennom innføringen av målingen. Følgelig er viktig å velge riktige KPI-er slik at man får frem den ønskelige arbeidsatferden.

3.2.2 Forskjellige planleggingsmetoder

Andersen (2018, p. 213) presenterer en visualisering av forskjellige planleggingsmodeller der man ser på mengde informasjon man har ved igangsetting av produktutvikling, slik som vist i Figur 3.3. Ved aktivitetsplanlegging, planlegges alt fra start, noe som kan være aktuelt «i tekniske prosjekter, der man har stor erfaring med tilsvarende leveranser.» (Andersen, 2018, p. 213). På motsatt side i figuren er agile metoder der man «hele tiden tar avgjørelser om hvilken leveranse prosjektet i neste omgang skal prioritere» (Andersen, 2018, p. 213). Midt imellom ligger milepælplanlegging der «man ved start lager en overordnet plan og supplerer den med detaljplaner etter hvert som arbeidet går framover og behovet for detaljplaner er til stede» (Andersen, 2018, p. 214).

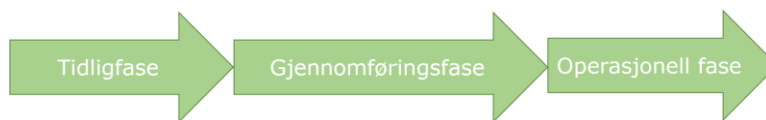


Figur 3.3: «Forskjellige planleggingsmodeller» (Andersen, 2018, p. 213).

3.2.3 Prosjekt

Samset (2015, p. 392) definerer et prosjekt som «et sett planlagte aktiviteter som har som formål å nå spesifikke mål innen en tidsfrist og et budsjett». Ifølge Yarbrough (2021), kan aktivitetene være rett frem eller mer komplekse avhengig av prosjektet, men «alle prosjekter kan deles inn i delmål og hva som trengs å gjøres for å nå dem» (Yarbrough, 2021). Samset (2015, p. 46) deler prosjekter inn i tre faser: tidligfase, gjennomføringsfase og operasjonell fase slik som illustrert i Figur 3.4. I tidligfasen er prosjektet konseptuelt og man har ikke laget noe fysisk enda. Denne fasen er fra da ideen først oppstod og til det er bekreftet at prosjektet

vil bli gjennomført. Gjennomføringsfasen inkluderer alle aktiviteter etter finansiering er bestemt og har mer detaljert prosjektplanlegging. Den operasjonelle fasen kommer etter prosjektet er ferdig og levert.



Figur 3.4: Visualisering av Samset sin faseinndeling av et prosjekt (Samset, 2015, pp. 46–47).

Et IT-prosjekt er et prosjekt der det utvikles «et nytt teknologibasert system eller service» (University of Mary Washington, 2016), det er tilrettelagt for «en signifikant transformasjon av forretningsprosesser ved bruk av teknologi» (University of Mary Washington, 2016) eller at det er inkludert «en stor endring i teknologisk arkitektur eller en systemmigring forbi det som anses som generelt vedlikehold, forbedring eller oppdateringsaktiviteter» (University of Mary Washington, 2016).

Liu (2016, p. 122) viser at mange IT-prosjekter leverer utilfredsstillende produkter til brukerne sine. «Bare 39% av prosjektene ble levert innen budsjettet, innen tidsfristen og med nødvendige funksjoner og egenskaper» (Liu, 2016, p. 122). Store prosjekter kan ha endringer i omfang, samt andre kvaliteter (Johansen et al., 2019, p. 5), så prosjektet trenger ikke være en suksess selv om man oppfyller kravspesifikasjonene som er satt på starten av prosjektet. Når utvikling blir målt på utdaterte kravspesifikasjoner, kan det bli vanskelig å utvikle dagsaktuelle produkter (Gothelf & Seiden, 2016, p. xx). Dette kan motvirkes gjennom kontinuerlig prioritering og detaljering av behov.

Når et prosjekt tar slutt, kan man miste de ressursene som er satt av til prosjektet. Eksempler på det er finansiering og personell. Når utviklingsoppgavene blir sendt til linja, kan arbeidsmetoden likevel være den samme, med smidig metode, sprinter og så videre, se kapittel 3.2.5. Utviklingsoppgavene overføres til linja og prioriteres av linjeledelsen opp mot andre linjeoppgaver. Oppgavene som er relatert til det som var i prosjektet kan dermed bli vurdert på lik linje som andre oppgaver. Hvis, eller når, prosjektet har nådd målene sine hender det at produktet ikke er ferdig enda. Utviklingshastigheten på produktet kan gå saktere når utviklingsoppgavene blir sendt til linja, nettopp fordi de blir vurdert på lik linje som andre oppgaver. Ved å ha en produktbasert utvikling istedenfor prosjektbasert, kan finansieringen være jevnere og man kan bli tildelt ressurser på en annen måte.

3.2.4 Fossefallsmetoden

Fossefallsmetoden er den mest tradisjonelle prosjektmetoden for store ikt-prosjekter. Ved bruk av denne metoden planlegger man alt i tidligfasen, før man beveger seg videre til gjennomføringsfasen (Andersen, 2018, p. 22). Deretter blir prosjektet overlevert, og produktet går inn i operasjonell fase. Figur 3.5 illustrerer dette. Ved fossefallsmetoden kan man ikke gå tilbake til tidligere faser, derav navnet fossefall der vannet kun renner nedover. Det kan være alt fra tre til syv faser når man benytter fossefallsmodellen (Andersen, 2018, p. 22). Med fossefallsmetoden settes kravene ved start, og man jobber for å innfri disse kravene. Dette kan føre til at prosjektet løser feil problem hvis behovene endrer seg

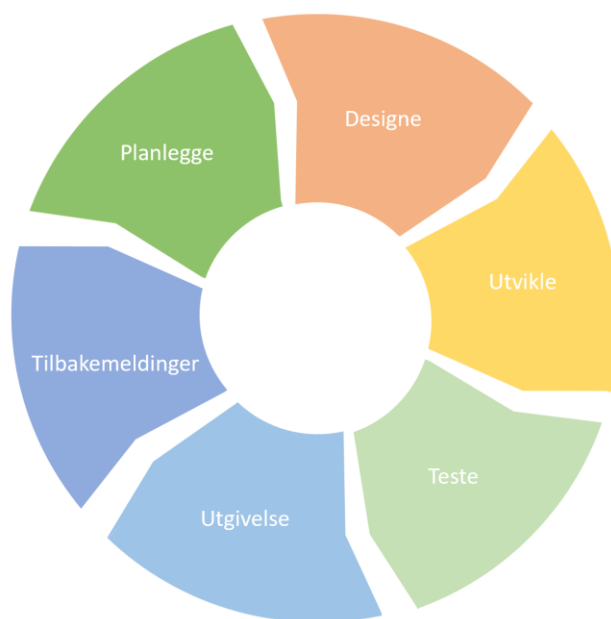
underveis, noe de som regel gjør. Det kan også være tilfellet hvis man ikke har fått god nok forståelse for hva behovene faktisk er før man begynner utviklingen.



Figur 3.5: Fossefallsmodell basert på Andersen (2018, p. 22) sin modell ⁶.

3.2.5 Smidig (agil) arbeidsmetode

Smidig metode er en svært populær metode når det kommer til softwareutvikling (Andersen, 2018, p. 256; Hoda et al., 2018, p. 58). Ved denne metoden jobber man iterativt, har trinnvise leveranser (Andersen, 2018, p. 256; Lamelas, 2018) og baserer seg på det smidige manifestet, the Agile Manifesto⁷, som er gjengitt i vedlegg D. Kontinuerlig integrering av løsning og kontinuerlig testing er viktige elementer i smidig utvikling (Meyer, 2018, p. 93). Dette kan sammenliknes med kontinuerlig eksperimentering som metode der man stadig eksperimenterer med verdier ved produktet, se kapittel 3.2.6. Ved å benytte smidige metoder kan utviklingsteamet lære gjennom prosessen, tilpasse seg til det de har lært og lettere løse opp i feil og problemer som oppstår underveis (Lamelas, 2018). Dette er illustrert i Figur 3.6.



Figur 3.6: Smidig arbeidsmetode (Adam, 2021; Admin, 2020; Introduction to Agile Methodology, 2020).

⁶ Eneste modifikasjon fra Andersen sin figur er at det er lagt til piler på enden av boksene for å få frem rekkefølge og avhengighet mellom fasene.

⁷ (Principles behind the Agile Manifesto, 2021)

I begynnelsen av et prosjekt vil ingen ha full forståelse av hva prosjektet vil oppnå og levere (Johansen et al., 2019, p. 54). Med smidige metoder er det lettere å tilpasse seg brukernes nye innsikt og se hva som funker og hva som ikke funker. Når man benytter smidige metoder er det et høyere fokus på å respondere kjapt enn å dokumentere godt (Saeeda et al., 2020, p. 2). Smidige metoder er nyttig når det bør være enkelt å endre kravspesifikasjoner (Saeeda et al., 2020, p. 2). En fordel med smidig arbeidsmetode er at man kan lære av hva som gir effekt og hva som ikke gir effekt underveis i utviklingsprosessen. Man oppdaterer mål og visjoner underveis som man lærer hvordan behovene endrer seg, eller at man tilegner seg mer kunnskap om hvordan situasjonen faktisk er. Saeeda et al. (2020, p. 1) sier at det finnes forskjellige hovedutfordringer ved kravspesifikasjoner når man jobber med smidig softwareutvikling. Noen av dem er å identifisere ufullstendige krav, implementering av vage krav, prioritering av sekundære funksjonelle krav og kompromisser på brukernes involvering som kan påvirke softwaresystemets funksjonalitet og kvalitet på en negativ måte (Saeeda et al., 2020, p. 1). Det finnes flere tilnærminger til smidig arbeidsmetode der en veldig populær tilnærming er gjennom Scrum (Adam, 2021; Hoda et al., 2018; Meyer, 2018). Da jobber teamet i iterasjoner som kalles «sprinter» (*What Is Scrum?*, 2021).

3.2.6 Kontinuerlig eksperimentering

Kontinuerlig eksperimentering er en fremgangsmåte som benyttes i Software utvikling der aktiviteter gjennomføres ved å kontinuerlig eksperimentere med verdier ved produktet (Yaman et al., 2016, p. 345). Ved å gjennomføre kontinuerlig eksperimentering kan man få en bedre forståelse av verdien av produktet og brukernes behov (Yaman et al., 2016, p. 345). Organisasjoner ser et behov for å endre forskning og utviklingsaktiviteter mot et eksperimentdrevet system som støtter kontinuerlig tilbakemeldinger fra brukere og mekanismer som fanger opp slike tilbakemeldinger (Yaman, 2018, p. 1). Gjennom prosjektene Yaman et al. (2017) observerte om å innføre kontinuerlig eksperimentering i større softwareorganisasjoner, fant de både fordeler og ulemper ved metoden. Fordeler var at man kan fatte beslutninger basert på empirisk data og man får ny innsikt med tanke på målene til virksomheten (Yaman et al., 2017, p. 37). Ulemper var at tidsfrister, kompliserte sammensetninger av interessenter og utilgjengelig tidligere data gjorde det vanskelig å gjennomføre de første eksperimentene, som videre gjorde det vanskelig å gjennomføre kontinuerlig eksperimentering (Yaman et al., 2017, p. 37).

3.2.7 Produktbasert utvikling

Softwareutvikling har blitt en kontinuerlig utviklingsprosess (Gothelf & Seiden, 2016, p. 3). Ved å benytte seg av smidige utviklingsmetoder og kontinuerlig implementering av nye løsninger kan utviklingsteamet redusere syklustid (Gothelf & Seiden, 2016, p. 4). Et eksempel er Amazon som gir ut ny kode til sine brukere hvert 11,6 sekund, noe som gir dem et konkurransefortrinn når det kommer til å benytte det de har lært fra brukerne sine (Gothelf & Seiden, 2016, p. 4). De oppdager produktet sitt samtidig som de leverer det til brukerne sine. Dette gir mange fordeler, der Gothelf og Seiden (2016, p. 4) anser de to viktigste å være:

- Muligheten til å lære, kontinuerlig og hurtig, hvor bra produktet når brukernes behov

- Heve brukernes forventninger i forbindelse med produktkvalitet og responstiden til virksomheten når det kommer til brukernes tilbakemeldinger.

Olsen (2015, p. xviii) introduserer konseptet produkt-marked-tilpasning (Product-market fit) der han forklarer hvordan fokuset bør være på denne relasjonen i utviklingen av nye produkter. Dette illustreres gjennom en pyramide slik som vist i Figur 3.7.

Videre har han laget en prosess der man guides gjennom pyramiden fra bunnen og opp. Prosessen består av seks steg (Olsen, 2015, p. xix):

1. Bestem hvem som er kundemålgruppa
2. Identifiser underleverte kundebehov
3. Definer ditt verdiforslag
4. Spesifiser egenskapene til ditt minimum viable product (MVP)
5. Lag en MVP-prototype
6. Test MVP med kundene



Figur 3.7: Norsk versjon av produkt-marked-tilpasningspyramiden (Olsen, 2015, p. xix).

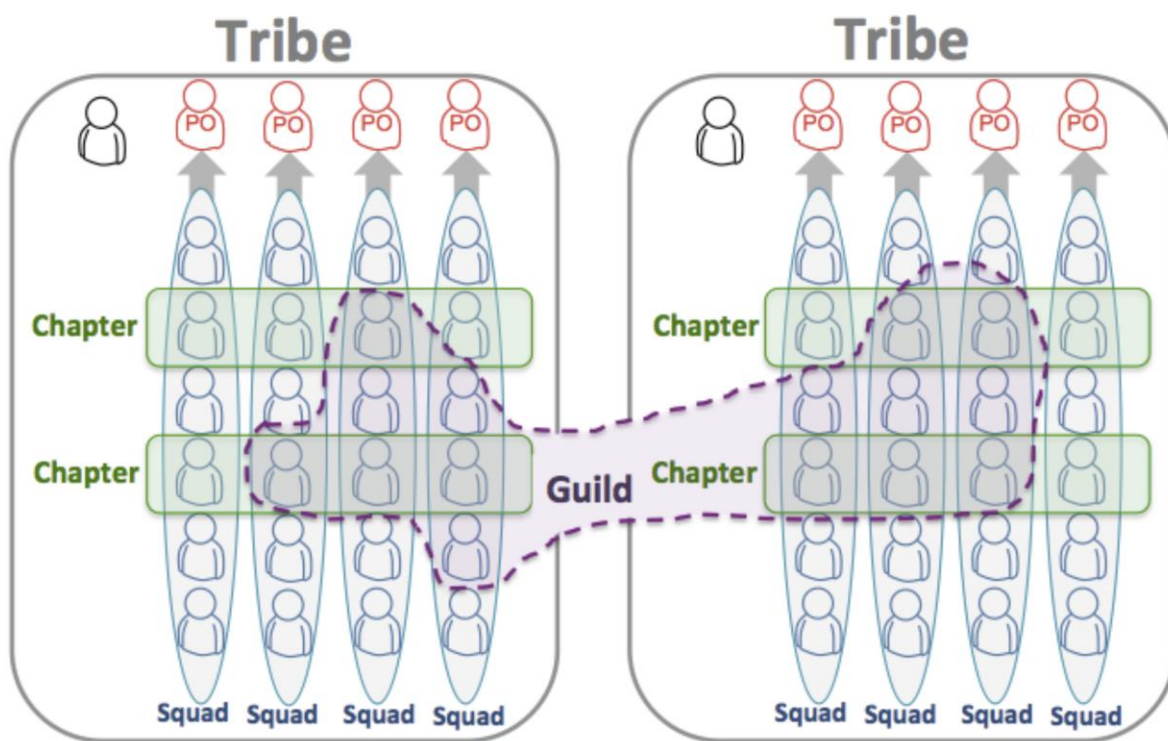
3.3 Team- og organisasjonsmodeller

3.3.1 Spotifymodellen

Spotifymodellen er en av flere organisasjonsmodeller som benyttes i storskala smidig utvikling (Dingsøyr et al., 2018, p. 10). Spotify benyttet en agil arbeidsmetode, selv om de er en veldig stor organisasjon med flere hundre utviklere (Kniberg & Ivarsson, 2012, p. 1). Spotify har gitt navn til en arbeidsmodell som deler inn organisasjonen i Squads, Tribes, Chapters og Guilds slik som vist i Figur 3.8. Hvert Squad jobber med hver sin leveranse til

Spotify (Kniberg & Ivarsson, 2012, p. 2). Ett Squad kan for eksempel jobbe med Spotify Radio, eller betalingsløsninger. Man blir oppfordret til å levere MVP (Minimum Viable Product), som er et produkt med nok funksjonalitet til å teste om konseptet fungerer. Man leverer funksjonalitet tidlig og ofte (Kniberg & Ivarsson, 2012, p. 2). Fra leveransene finner man ut av hva som gir verdi og ikke, gjennom for eksempel A/B testing. A/B testing er at man gir ut to versjoner av en løsning og observerer hvilken som fungerer best⁸.

Modellen bygger opp organisasjonen som en type matrise, men med vektning mer mot leveranse enn mot funksjonelle avdelinger (Kniberg & Ivarsson, 2012, p. 11). En Tribe inneholder flere Squads som jobber på relaterte områder (Kniberg & Ivarsson, 2012, p. 5). Squads kan også samarbeide eller være avhengige av andre Squads på tvers av Tribes. Et Chapter består av folk som har samme kompetanse/arbeidsområde innenfor en Tribe (Kniberg & Ivarsson, 2012, p. 9). Eksempler på Chapters er testing og webutvikling. Videre har man Guilds. En Guild kan sammenliknes med et Chapter, men en Guild trenger ikke holde seg innenfor en Tribe og kan bre seg ut over hele organisasjonen (Kniberg & Ivarsson, 2012, p. 10). Eksempler på Guilds er webteknologi og agile trenere. Spotifymodellen idealiseres av mange virksomheter, men Spotify benytter den ikke selv (Lee, 2022), noe som er viktig å huske på når man streber etter å følge «deres» fremgangsmåte.



Figur 3.8: Spotifymodellen (Kniberg & Ivarsson, 2012, p. 1).

⁸ (A/B testing, 2022)

3.3.2 Extreme programming (XP)

Extreme programming (XP) er en teknikk og en filosofi som benyttes i softwareutvikling (Beck & Andres, 2005, pp. 1–2) og er fra tidlig 2000-tallet, men blir fortsatt brukt i dag. Den benytter kode som hovedaktivitet gjennom softwareutvikling (Beck & Andres, 2005, p. xix). XP baserer seg på de fem verdiene kommunikasjon, tilbakemelding, enkelhet, mot og respekt (Agile Alliance, 2017; Beck & Andres, 2005, p. 2). Det handler om å kun gjøre det man absolutt må for å gi verdi til kunden (Beck & Andres, 2005, p. 3). Forskjellige team gjennomfører metoden på forskjellige måter med varierende grad av suksess (Beck & Andres, 2005, p. 3). God og trygg sosial interaksjon er vel så viktig som gode tekniske ferdigheter for å oppnå suksess ved bruk av XP (Beck & Andres, 2005, p. 4). Metoden har noen aktiviteter som kan gjennomføres alene, men flere team har erfart at aktivitetene styrker hverandre ved å gjennomføre dem sammen (Agile Alliance, 2017). Beck og Andres (2005, pp. 6–7) gir en kortfattet punktliste om hva XP handler om:

- Gi opp gamle, ineffektive tekniske og sosiale vaner til fordel for nye som fungerer
- Være takknemlig for egen innsats gjennom dagen
- Strebe etter å gjøre en bedre jobb i morgen
- Evaluere deg selv basert på ditt bidrag til teamets delte mål
- Spørre om at noen av dine menneskelige behov blir møtt gjennom software utvikling

Når man benytter XP-prinsipper, så er ikke rollene fastsatt og strenge og man har et mål om at alle skal bidra så godt de kan for teamets suksess (Beck & Andres, 2005, p. 82).

4 NAV som caseorganisasjon

Kapittel 4.1 og 4.2 gir et innblikk i NAV som organisasjon og prosessen for søknad om sykepenger og er basert på NAV sine og regjeringen sine hjemmesider. Kapittel 4.3, 4.4 og 4.5 gjengir tolkning av arbeidssituasjonen og er innhentet gjennom intervjuer og samtaler med teammedlemmer.

4.1 NAV som organisasjon

I 2006 ble NAV etablert og har i dag ca. 20 000 ansatte der ca. 15 000 er i staten og ca. 5000 er i kommunene⁹. Statlig og kommunalt ansatte i NAV jobber sammen om å finne de beste løsningene for brukerne sine¹⁰. «Arbeids- og velferdsetaten (NAV) skal bidra til å skape et inkluderende samfunn, et inkluderende arbeidsliv og et velfungerende arbeidsmarked. Etaten skal ivareta vanskeligstilte gruppers behov og bekjempe fattigdom [...]»¹¹. Årlig betjener NAV 2,8 millioner brukere og de forvalter ca en tredjedel av statsbudsjettet¹¹. NAV har følgende hovedmål, som kan finnes på deres nettsider⁹:

- Flere i arbeid og aktivitet, færre på støtte
- Et velfungerende arbeidsmarked
- Rett tjeneste og støtte til rett tid
- God service tilpasset brukeren sine forutsetninger og behov
- En helhetlig og effektiv arbeids- og velferdsforvaltning.

4.2 Prosessen for søknad om sykepenger

Hvis en arbeidstaker blir syk har man rett på sykepenger hvis man er innenfor de forenklede hovedreglene som er gjengitt i følgende kulepunktliste¹². Ellers forholder man seg til kapittel 8 i Folketryktdloven i behandling av sykepengesaker.

- Er medlem av folketrygden
- Er under 70 år
- Er syk eller skadet
- Har vært i jobb i minst fire uker umiddelbart før man ble sykmeldt
- Må være borte fra jobb minst 20% på grunn av sykefraværet
- Inntekten tilsvare minst 50% av Folketrygdens grunnbeløp, der grunnbeløpet per 1. mai 2021 er 106 399 kr¹³.

Når man har blitt syk, må man enten levere egenmelding til arbeidsgiver eller sykemelding til arbeidsgiver og NAV¹². Deretter må man søke om sykepenger fra NAV. Selv om man får lønn fra arbeidsgiver mens man er syk, må man søke om sykepenger fra NAV¹². Det er fordi

⁹ (Kva er NAV?, 2022)

¹⁰ (Organisering av NAV, 2022)

¹¹ (inkluderingsdepartementet, 2006)

¹² (Sykepenger til arbeidstakere, 2022)

¹³ (Grunnbeløpet i folketrygden, 2021)

arbeidsgiveren «krever pengene tilbake fra NAV basert på din søknad»¹². Arbeidsgiver må også sende inn inntektsmelding, hvor det blant annet sendes inn inntektsopplysninger.

Etter at søknaden for sykepenger er mottatt av NAV må den behandles. I dag gjøres det av saksbehandlere. De behandler saker manuelt ved å behandle søknaden og inntektsmeldingen og vurdere hvordan det henger sammen opp mot lovverket. Som saksbehandler jobber man med å utrede og besvare søknader, notater og andre skriv¹⁴. Ifølge en stillingsutlysning fra NAV, er arbeidsoppgavene til en saksbehandler «saksbehandling og utbetaling av arbeids- og velferdsytelser i henhold til gjeldende regelverk. Bidra til kvalitet og måloppnåelse av enhetens mål.»¹⁵. De utbetaler sykepenger som et resultat av søknadsbehandlingen, eller gir avslag med begrunnelse i lovverket. Saksbehandlere trenger velfungerende systemer for å behandle saker. Utviklingsteamet som studeres i denne masteroppgaven, jobber med å erstatte et gammelt saksbehandlingssystem med en ny og moderne løsning, som automatiserer store deler av saksbehandlingsprosessen.

4.3 Utviklingsteamets oppbygging og ansvarsområder

Utviklingsteamet kaller seg Team Bømlo og består av to produkteiere, tre jurister, to designere, en data scientist, en teamleder og ca. 16 utviklere. Figur 4.1 viser teamets oppbygging. Teammedlemmene er de grønne boblene nærmest kjernen. De blå boblene lenger ute er støttefunksjoner som teamet kan trekke på når det trengs. Da studiet av teamet ble gjennomført var teamet delt inn i tre subteam. Det ene subteamet jobbet med fortløpende feilretting, videreutviklet eksisterende funksjonalitet og dekte oppgavene som er en del av kontinuerlig produktutvikling. De to andre subteamene utviklet ny funksjonalitet og fordelte oppgaver seg imellom ut ifra hva som var fokuset i teamet og hvilke mål som skulle nås. Målene og fokuset har direkte påvirket teamstrukturen, og denne strukturen blir omgjort etter behov.

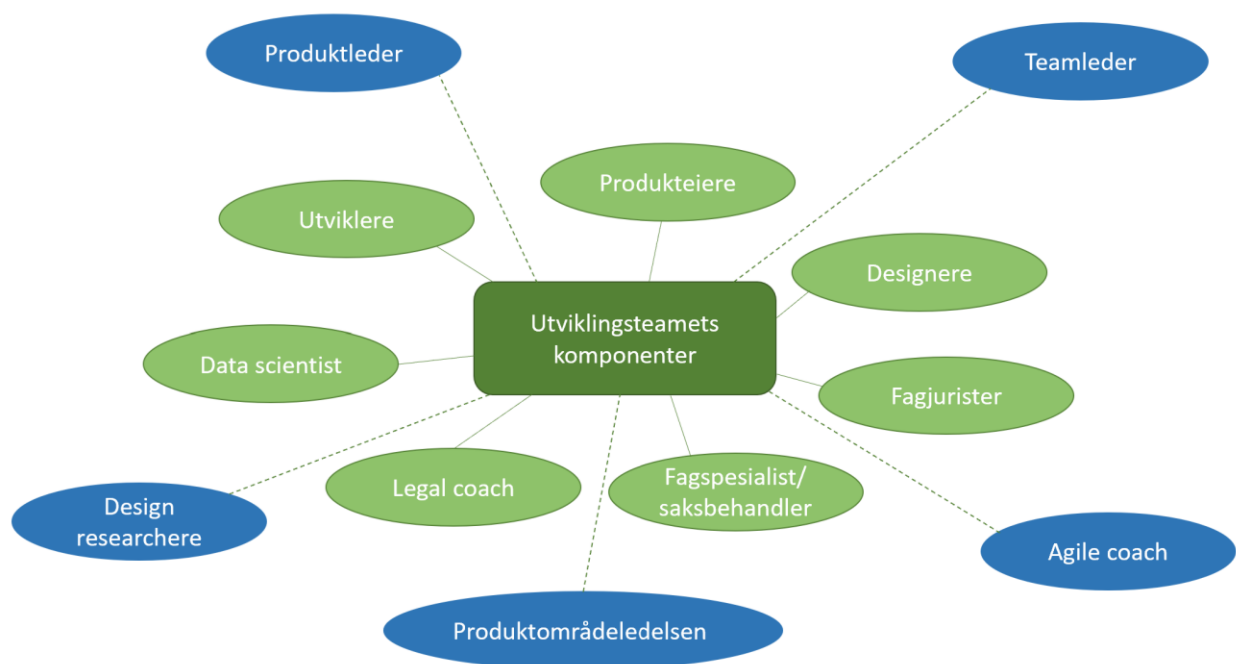
Brukermedvirkning defineres av utviklingsteammedlemmer som aktiviteter der brukerne er med og påvirker hvordan løsningen skal bli. Videre definerer de smidig metode som at man produserer små ting der man lærer av det man har laget og tar det med videre i utviklingsprosessen. Hovedformålet med smidig er at man ikke bruker så mye tid på å planlegge, man bruker ikke så mye tid på forberedende aktiviteter, men er heller mer opptatt av å lære underveis. Hvordan man lærer underveis er veldig åpent. Det er enighet om at hoved essensen til smidig metode er at det er læring hele tiden som gjør at man kan bli bedre, både på prosessen og på produktet. Utviklingsteamet har et fokus på å se bort fra klare roller innad i teamet.

Etter at covid-19 slo til har teamet blitt så og si heldigitale («Remote first»). De kommuniserer for det meste over Slack, Zoom og Teams. Kommunikasjonen over disse kanalene er innad i teamet, med andre team og enheter i NAV og med saksbehandlerne. Teams brukes i hovedsak til kommunikasjon med andre deler av organisasjonen. Brukerne av systemet, altså saksbehandlerne, befinner seg i hele landet. Nå kan teamet kjapt ta kontakt med brukerne

¹⁴ (Saksbehandler, 2022)

¹⁵ (Ledige stillinger - Arbeidsplassen, 2022)

gjennom disse digitale verktøyene. Informasjonen som innhentes gjennom brukermedvirkningsaktiviteter lagres gjennom anonymiserte sammendrag i Slack, Trello, skisser, Excel-dokumenter, PowerPoint i Teams og i Sharepoint.



Figur 4.1: Teamets oppbygging

Designerne driver saksbehandlertreffene og testing/tilbakemeldinger på skisser og tekst. De har som regel flere møter i løpet av uka. Møtene gjelder ofte koordinering med andre team og med andre designere.

Utviklerne bruker mye tid på utviklingsoppgaver. Det er et rullerende ansvar hos utviklerne der én utvikler hele tiden har ansvar om å følge opp taggen «red team» i Slack. Det vil si at hvis «red team» tagges i Slack så tror vedkommende som tagger at dette er en teknisk feil. Da har én utvikler ansvar for å få analysere problemet og vurdere hva som burde gjøres videre. Utviklerne trenger konsentrasjon for å følge opp lange tankerekker der det å være på Slack vil gi brudd på konsentrasjonen. Teammedlemmer kan hoppe inn i utviklerne sitt Zoom-rom og be om hjelp eller tilkalle utviklerne til andre Zoom-rom for å se på noe man trenger hjelp med umiddelbart. Utviklerne oppdaterer brukerne via Slack når de har fikset noe. Dette er for at saksbehandlerne skal kunne holde seg oppdatert og at de skal se at teamet faktisk utvikler det som er etterspurt. Utviklerne, og resten av teamet, driver med testdrevet utvikling¹⁶ der teamet tester om funksjonaliteten fungerer som tiltenkt i et testmiljø.

Produkteierne jobber ganske likt som designerne, men har en del mer koordineringsmøter inn mot andre team. De er også med på alle de faste møtene med saksbehandlerne.

¹⁶ Tilnærmet «softwaretester» slik som forklart i kapittel 3.1.6

Produktlederen har en litt diffus rolle inn mot teamet. Vedkommende er med på ledernivå og snakker mye med avdelingsledere og enhetsledere. Vedkommende har også mer strategisk ansvar for hva som skjer rundt i organisasjonen som påvirker teamet. Teamet har en datascientist som jobber med analyse av data som samles inn. Et eksempel på data er antall saker som ikke kan behandles av det automatiske systemet, og hvorfor sakene blir kastet ut av den automatiske løsningen. Juristene har mye møter, men er også noe på Slack. Samtlige teammedlemmer er med i rulleringen om å være med på saksbehandlertreff. Teamet har en Agil Coach som jobber mer på utsiden av teamet og er rettet mer mot prosess. Den Agile Coachen har ikke vært med på noe med tanke på brukervedvirkning. Teamet har også fått inn en ny teamleder som jobber veldig politisk rundt i organisasjonen.

Ansvar for andre aktiviteter som kan ligge under brukervedvirkning, som faste møtepunkter, Slack-aktivitet og å svare på spørsmål, ligger på de som eier innkallingen eller at det skjer av seg selv gjennom Slack. Hele utviklingsprosessen til teamet er på mange vis brukersentrert.

4.4 Utviklingsteamets arbeidsmetoder

Team Bømlo ligger et sted mellom milepælplanlegging og agile metoder slik som presentert i Figur 3.3. De jobber smidig, rettet mot produktbasert utvikling, og benytter en blanding av Spotify-modellen, XP-programmering og kontinuerlig eksperimentering. De benytter seg lite av konstruert brukertesting (se kapittel 3.1.6), da de heller observerer hvordan produktet faktisk blir tatt i bruk. Dette er for å lære hvordan systemet fungerer i virkelige settinger.

Teamet benytter seg av syv prinsipper for hvordan de ønsker å jobbe som et produktfokustert utviklingsteam. Disse kan sees i vedlegg E. Prinsippene omhandler arbeidsmetode gjennom samhandling med brukerne, organisering av teamet og utvikling av produktet.

Teamet benytter seg mye av testdrevet utvikling der de tester at koden gjør det som er tiltenkt. De benytter seg også av verktøy for monitorering av avvik. Dette er i henhold til et av disse syv prinsippene der man skal teste, måle og analysere resultatet. Softwaretestene gjennomføres ved at oppgaver som skal prioriteres, og er klare for utvikling, legges i arbeidsflytverktøyet som benyttes, Trello, i form av en lapp i kolonnen som de har kalt «Ready to play». Når utviklerne jobber med oppgaven, flytter de lappen til kolonnen de har kalt «Doing». Når utviklerne anser oppgaven som ferdig, plasserer de lappen i «Ready to review»-kolonnen. Alle på teamet kan teste funksjonalitet til lappene som er plassert i denne kolonnen, avhengig av hva slags funksjonalitet det er snakk om. For eksempel vil utviklerne teste rent tekniske aspekter, mens jurister vil teste juridiske aspekter. Man vil typisk teste funksjonaliteten som lappen beskriver i testmiljøet for å se om det ble som tiltenkt. Man prøver også å finne ut om det er situasjoner man ikke har tenkt på, eller situasjoner der det er mulig for brukeren å gjøre feil.

Utviklingsteamet jobber stort sett digitalt. Arbeidsdagene gjennomføres for det meste på Zoom der de har forskjellige «grupperom» som man kan hoppe mellom ut ifra hvem man skal jobbe med. Der har de ett fellesrom og de forskjellige subteamene har hvert sitt rom. I tillegg har de andre digitale rom i Zoom som kan benyttes hvis man skal møtes på tvers av subteam. Når det gjennomføres parprogrammering benyttes disse andre rommene. Flere teammedlemmer jobber selvstendig før og mellom Zoom-møter. Slack benyttes som skriftlig

kommunikasjonskanal og alle leser det som legges ut der. Mye kommunikasjon mellom utviklingsteammedlemmer skjer også gjennom Slack. Innimellom oppstår det «branner», både store og små, gjennom dagen som slokkes litt som de kommer.

Fra og med 1. mars er det 25 saksbehandlere som jobber med manuell saksbehandling i ny løsning til daglig. Før dette var det så mange som 200 som jobbet i denne fra tid til annen. Det kunne gå flere uker mellom hver gang saksbehandlerne var innom, noe som gjorde det vanskelig både for utviklingsteamet og for saksbehandlerne siden det skjedde så mange endringer i systemet mellom hver gang. Derfor gikk de over til å være færre saksbehandlere som jobber i løsningen til daglig.

«Den ene tingen vi har savnet, som har blitt bedre nå, er å ha færre saksbehandlere og forholde oss til. Strategien var lenge at alle disse sykepengeenhetene skulle læres opp i ny løsning. Da ble det veldig spredt og det var mange som var innom ny løsning innimellom. [...] Så det å få tyngre brukere, altså brukere som oftere bruker produktet og gjør det mer kontinuerlig, tror jeg er nyttig både for saksbehandlerne og for oss.»

– Teammedlem

Ett av de syv prinsippene sier at man skal sette seg inn i brukerens situasjon. Utviklingsteamet jobber tett med de 25 saksbehandlerne som arbeider i systemet til daglig. Fem av disse har jobbet i løsningen fra da utviklingen begynte, og kalles supersaksbehandlere. De ble valgt for å representere ulike enheter av saksbehandlere, men det er ikke dokumentert hvordan de har blitt valgt. Supersaksbehandlerne saksbehandler og gjennomfører opplæring av saksbehandlere. «Vanlige» saksbehandlere skal kontakte supersaksbehandlerne først hvis de lurer på noe. Supersaksbehandlerne kan deretter henvise saksbehandleren til noen i utviklingsteamet, eller si at de kan legge ut en melding i Slack om det de lurer på.

Team Bømlo gjennomfører en rekke brukermedvirkningsaktiviteter med sine brukere, der forskjellige roller deltar i forskjellig grad. De har ukentlige møter med supersaksbehandlere der saksbehandlere lufter problemer og forteller hva de synes er vanskelig om dagen. Teamet har også ukentlige møter med alle saksbehandlerne som jobber i løsningen. De lager også spørreundersøkelser, både for å grave mer i spesifikke ting og for å sjekke hva utviklingsteamet kan bli bedre på. Kommunikasjonsplattformen Slack benyttes aktivt og utviklingsteamet svarer hyppig på henvendelser der. De informerer saksbehandlerne via Slack når de introduserer ny funksjonalitet. De deler også skisser med saksbehandlerne og får tilbakemelding på disse. De svarer på support og andre spørsmål fra saksbehandlerne. Den største brukermedvirkningsaktiviteten de gjennomfører er observasjon av saksbehandlerne i arbeid, som kalles saksbehandlertreff. Dette kan leses mer om i kapittel 5.6.

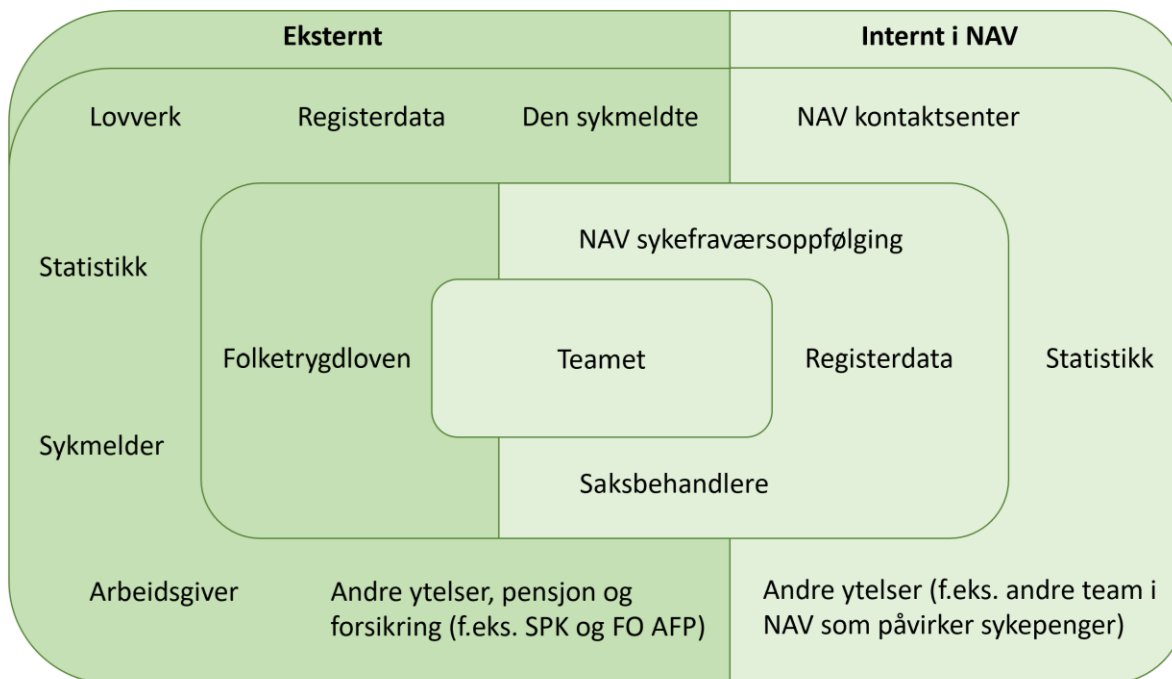
I Slack deles også skisser og spørsmål til og fra saksbehandlere for å få tilbakemeldinger og for å få og gi svar. Både utviklingsteamet og saksbehandlere benytter plattformen for å kommunisere med hverandre. Her lanseres også ny funksjonalitet. Det har blitt innført et taggesystem der man tagger forskjellige team avhengig av hvem som kan hjelpe med å løse problemet. Dette skal gjøre det enklere å komme frem til de riktige personene for det

gjeldende problemet eller spørsmålet. Det benyttes emojis og internhumor som har som hensikt å bygge en mer personlig tone.

Det jobbes *ikke* med kravspesifikasjoner. Istedenfor har utviklingsteamet fokusområder der man samarbeider og snakker sammen som et tverrfaglig team for å finne ut hvordan problemstillingen skal løses. Brukerne har ikke krav i den forstand at de kan framlegge et krav som utviklingsteamet må dekke. Brukerne kan komme med ønsker som teamet kan vurdere. Ofte vil et ønske bearbeides noe, der det puttes i en større sammenheng og veies opp mot andre ting, før det vedtas noe endelig. Dette dokumenteres gjennom brukerhistorier og oppgaver som legges inn som lapper i Trello.

Overordnede mål for teamet er for eksempel å redusere antall saksbehandlere som trengs for å håndtere sykepengesakene i Norge, komme ut av gammel saksbehandlingsløsning og å bidra til en bedre brukeropplevelse for de som er syke i Norge. Brutt ned ser de på dataen for «hva skal til for å nå overordnede mål». For eksempel kan et fokus være å få på plass funksjonalitet for utbetaling direkte til den sykmeldte fordi de ser av dataene de har at dette er den største gruppa med oppgaver som per nå ikke kan behandles i ny løsning. Dermed benytter teamet noe datadrevet beslutningsstøtte. For å få på plass utbetaling til sykmeldt, har de jobbet med å bryte ned «hva skal til for å få det på plass» som har vært gjort av teamet, der juristene for eksempel spiller en rolle sammen med ledere, saksbehandlere og ledelsen til saksbehandlerne. Under der igjen er mange større oppgaver som de bryter ned til mindre steg og oppgaver. Dette må prioriteres opp mot hva som skal til for å nå fokuset, hva slags hendelser som skjer i produksjon (også kalt drift) og hva slags andre hendelser det er som må løses som skjer hvert år, for eksempel grunnbeløpsendring og feriepengekjøring. Saksbehandlerne er ikke direkte med på å sette disse målene, men utviklingsteamet lærer av dem, noe som påvirker prioriteringene i utviklingen.

Teamet har flere interne og eksterne avhengigheter. Dette er fremstilt i Figur 4.2, som er en forenklet figur for å vise kompleksiteten. Hvis alle detaljer skulle vært med, ville figuren blitt for stor og komplisert. Partene i den høyre delen er interne i NAV og de til venstre er eksterne. Videre er partene delt inn i primære og sekundære påvirkninger, der de nærmest teamet er primære og de ytterste er sekundære.

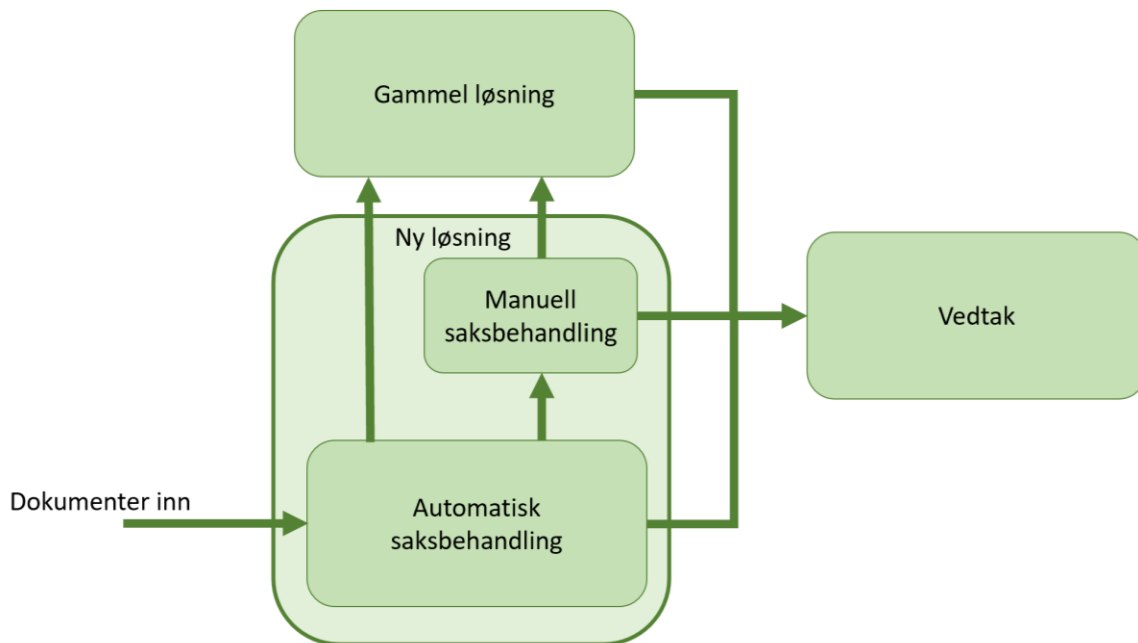


Figur 4.2: Parter som teamet påvirker og som påvirker teamet.

4.5 Produktet

Produktet som teamet utvikler, er en løsning for saksbehandling av sykepenger. Det jobbes i stor grad med å legge til ny funksjonalitet for å ta flere saker over fra gammel til ny løsning. Løsningen skal automatisere saksbehandling og manuelt behandle saker som krever det. Sistnevnte kan være saker som virker mistenkelige med tanke på svindel eller liknende. Det kan også være uryddige register som må sjekkes manuelt eller at man må kontakte arbeidsgiver for å klarere ting. Det kan også være saker som det ikke lønner seg å behandle automatisk fordi det blir for komplisert kode. Saker kan kreve skjønnsmessig vurdering av saksbehandler og må følgelig behandles manuelt. Figur 4.3 visualiserer sammenheng mellom ny og gammel løsning, og hvordan de forskjellige delene fungerer sammen. Alle dokumenter blir sendt inn til den automatiske saksbehandlingen i ny løsning. De sakene som ikke kan behandles automatisk blir sendt til manuell behandling i den nye løsningen eller direkte til den gamle løsningen. Dokumenter som har blitt sendt til manuell behandling i den nye løsningen, men som det mangler funksjonalitet til å løse sendes til gammel løsning for å behandles der. Alle saker blir sendt videre gjennom et vedtak der saken enten er blitt godkjent eller blitt avslått.

To av de syv prinsippene som teamet jobber etter er å levere verdi ofte og å realisere produktområdets mål. Dette etterleves ved å ha denne kombinasjonen mellom ny og gammel løsning slik at produktområdets mål hele tiden kan overholdes samtidig som det nye produktet er i utvikling. Det blir levert verdi ofte gjennom å ha hyppige lanseringer som inneholder ny funksjonalitet og forbedringer av tidligere lansert funksjonalitet.



Figur 4.3: Forenklet visualisering av produktet som leveres fra utviklingsteamet.

5 Resultater

Dette kapitlet er delt inn i temaene teamet og organisering, holdninger og motivasjon til brukermedvirkningsaktiviteter, kommunikasjon, dokumentasjon, supersaksbehandlerrollen, saksbehandlertreff og finansiering av utviklingen. En oppsummering av funnene er gjengitt i Tabell 5.1.

Tema	Funn
Teamet og organisering	<ul style="list-style-type: none"> • Teamet benytter seg av en prøv-og-feil-metode. • Digitaliseringen av teamet i sammenheng med covid-19 oppleves som positiv. • I startprosessen var teamet utforskende for å lære hvordan produktet burde bli. • Læringsloopen fungerer ikke optimalt. • Ett teammedlem utmerkes som veldig flink til å involvere alle i teamet med hverandre. • Behov for bedre ansvarsavklaring når det kommer til arbeidsmetode, utviklingens mål, fremdrift og milepæler. • Saksbehandler ønsker tidligere involvering. • Stort arbeidspress både hos saksbehandlere og hos utviklingsteamet noe som kan skyldes måten prestasjoner måles. • Saksbehandling blir målt på antall saker man klarer å saksbehandle og det er for få som jobber med behandling av sykepengesaker i dagens situasjon. • Det oppleves som positivt å ha færre brukere av systemet og det kan gi et bedre samarbeid mellom partene. • Konstruerte brukertester anses som oppkonstruerte av teammedlemmer. • Et teammedlem opplever det som at det er for lite forståelse i organisasjonen for viktigheten av brukermedvirkning i riggen av sykepengesaker.
Holdninger og motivasjon til brukermedvirkningsaktiviteter	<ul style="list-style-type: none"> • Varierende motivasjon for brukermedvirkning i organisasjonen; i Team Bømlo oppleves det som god forståelse for viktigheten av brukermedvirkning. • Brukerne blir hensyntatt i stor grad. • Klikkbare prototyper og konstruerte brukertester benyttes i begrenset grad fordi det anses som vanskelig å finne ut om man får testet de riktige tingene i en slik test. • Gjennom arbeidsdagen blir brukernes innspill tatt hensyn til i stor grad.
Kommunikasjon	<ul style="list-style-type: none"> • Det observeres en lett og saklig kommunikasjon. • Misforståelser oppleves av begge parter å bli løst på en god måte.

	<ul style="list-style-type: none"> • En utfordring å kommunisere tekniske egenskaper på et passende nivå fra utviklingsteammedlem til saksbehandler. • Kalle det «fagpersoner som diskuterer en problemstilling» istedenfor «brukermedvirkning». • Kommunikasjonen utenom saksbehandlertreffene må være god. • En utfordring at utviklingsteamet ikke skjønner hvorfor noe er viktig for saksbehandlerne; en utfordring å kommunisere utviklingsteamets prioriteringer. • Tett samarbeid med teamet gir saksbehandlere bedre forståelse av systemet. • Automatiseringen av behandling av saker har ikke blitt gjort fort nok i forhold til hva saksbehandlere har blitt lovet. • Forslag fra et teammedlem om implementering av et finkornet tilbakemeldingsverktøy. • Kontinuerlig dialog om planer, nåværende oppgaver og fremtidige prioriteringer må bli bedre. <p><u>Kommunikasjon gjennom digital skriftlig kommunikasjonskanal, Slack:</u></p> <ul style="list-style-type: none"> • Uenighet mellom saksbehandlere og teammedlemmer om hvor lav terskelen er for å legge ut ting på Slack. • Lansering av ny funksjonalitet gjøres på en ryddig måte via Slack. • Det er ikke alltid partene forstår den andres vurdering av viktigheten til et problem via Slack. • Generelt god og tydelig kommunikasjon på Slack.
Dokumentasjon	<ul style="list-style-type: none"> • Det er et ønske fra saksbehandlere om mer utfyllende dokumentasjon om fremdrift og hva som er gjeldende i systemet. • Koden til systemet har versjonshåndtering slik at man kan se hva som var gjeldende funksjonalitet på et tidligere tidspunkt. • Teammedlemmer har oppdaget en feil i en ordlyd.
Super-saksbehandlerrollen	<ul style="list-style-type: none"> • Representasjonen fra enhetene i NAV som er mest ulike er for dårlig i gruppa med supersaksbehandlere. • Det er usikkerhet fra en saksbehandler rundt hva som er supersaksbehandlerne sitt ansvarsområde. • Det oppleves frustrasjon rundt at «alt» må gjennom supersaksbehandlerne. • Supersaksbehandlere og vanlige saksbehandlere deler forskjellige ting med teamet, trolig fordi man blir kjent med dem i forskjellig grad.
Saksbehandler-treff	<ul style="list-style-type: none"> • Saksbehandlertreff oppleves av alle intervjuobjekter å være nyttige. • Mindre formelle saksbehandlertreff benyttes også. • Rent juridiske feil som oppdages under treffene blir tatt opp i etterkant av treffet.

	<ul style="list-style-type: none"> • «Gamle» feil i systemet kan oppdages at fortsatt eksisterer under treffene. • Saksbehandlere føler det er viktig å benytte denne sjansen til å vise og forklare hva de trenger av løsningen. • Kommunikasjonen mellom partene oppleves god fra begge sider. • En frykt fra utviklingsteamet at saksbehandlere skal føle seg «testet» under treffene. • Et personlig forhold til brukerne av systemet gir en annen motivasjon. • Treffene hjelper saksbehandlerne med å skjønne systemet, noe som gir dem mer trygghet i systemets vurderinger. • Treffene «tvinger» teammedlemmene til å se ting i et større perspektiv.
Finansiering av utviklingen	<ul style="list-style-type: none"> • Et teammedlem ønsker en annen finansieringsmetode for utvikling av IT-produkter

Tabell 5.1: Oppsummering av funn

5.1 Teamet og organisering

Ifølge et utviklingsteammedlem innebærer arbeidsmetoden til teamet at det aksepteres at feil kan skje, men at de må oppdage feilene raskest mulig. I tillegg til teknisk overvåking, monitorering og metrikker, så betyr det at saksbehandlerne kontakter teamet hvis de finner feil. Vedkommende påpeker at å prøve og feile anses som et viktig bidrag til læring. Å ha muligheten til å se hvordan programvaren tas i bruk i virkeligheten gir enorm verdi. En utfordring er ofte at det er vanskelig å se for seg alle mulige scenarier som kan oppstå, både tekniske og funksjonelle, før man støter på reelle data. En mentalitet i utviklingsteamets arbeidsform er at for å lære raskest mulig, må man ha ting i produksjon raskest mulig fordi det er der man lærer om det feiler. Statistikkpakker benyttes for å se på hvordan funksjonalitet treffer brukerne.

Gjennomføringen av å digitalisere teamet da covid-19 kom anses som positiv av saksbehandlere. En saksbehandler opplever det som at utviklingsteamet har tillagt saksbehandlerne vaner ved hjemmekontoret som anses å ha gjort det lettere med hjemmekontorsituasjonen. Et utviklingsteammedlem og en saksbehandler opplever at digitaliseringen har gjort kommunikasjonsprosessen litt vanskeligere, men at det også har gitt verktøy som har gjort prosesser bedre og lettere for saksbehandlerne.

Agendaen for utviklingsteammedlemmenes arbeidsdag observeres som å være delvis klar. Gjennom dagen blir det gjort løpende vurderinger om gjennomføring av aktiviteter fortsatt er relevant og hensiktsmessig å gjennomføre.

I startprosessen var utviklingsteamet utforskende der de snakket med andre som også utviklet saksbehandlingssystemer, noe de også gjør kontinuerlig, og ble kjent med lovverket. Utviklingsteammedlemmer reiste rundt til saksbehandlerne sine arbeidsplasser. Der gjennomførte de observasjoner av saksbehandlere i arbeid i gammel løsning og dybdeintervjuer med saksbehandlerne. Under observasjonene prøvde de å forstå hva saksbehandlerne faktisk gjør, og hvilke støtteverktøy de har. Det som gjøres på skjermen under saksbehandling er én ting, men det som gjøres utenfor skjermen viser hva løsningen

ikke dekker. De gjennomførte reiser hver uke eller annenhver uke. De hadde dette program for hver reise og passet på å ha oppdaterte prototyper til hver gang de skulle reise. De lagde klikkbare prototyper som blant annet juristene kunne sitte og trykke på for å skjønne systemet riktig og sørge for at systemet holdt seg innenfor jussen. Faktiske søknader kan ikke benyttes uten at de saksbehandles, så hvis de skulle benytte en søknad til å teste løsningen måtte den bli en del av saksbehandlingen. På dette tidspunktet hadde de ikke utbetalingen på plass, så systemet ble benyttet som et støtteverktøy og utviklingsteammedlemmer satt i samme rom som saksbehandleren og observerte behandlingen. De benyttet lærdommen fra dette til å bygge ut løsningen.

En saksbehandler savner et bedre forarbeid ved å bli involvert i oppstartsprosessen. Ved å involvere saksbehandlere tidligere og gi god innføring i prosjektet og målsettingen, kunne man spare mye tid der saksbehandlere lurer på ting. Det oppleves av denne saksbehandleren at de er blitt utelatt av verdikjeden. Likevel er det forståelse for at det kan være vanskelig å ha for mange mennesker med i en tidlig fase.

Et utviklingsteammedlem anser den største utfordringen med å jobbe smidig å være at man snakker om å jobbe smidig, men at læringsloopen der man går tilbake og benytter ting man har lært ikke fungerer optimalt. Dette mener vedkommende er fordi teamet hele tiden blir presset fremover for å levere ny funksjonalitet. Teammedlemmet påpeker også at det er viktig for utviklingsteamet at de følger opp forslag til forbedring fra saksbehandlerne. Dette anses som en del av det å bygge tillit.

Flere teammedlemmer nevner at det ene teammedlemmet er veldig flink til å involvere alle i teamet med hverandre. Vedkommende er veldig flink til å bygge broer og se sammenhenger som andre ikke helt har kapasitet til. Designerne blir også påpekt som flinke av saksbehandlere når det kommer til involvering av brukere.

«[Det ene teammedlemmet] ... er veldig flink til å involvere alle i teamet med hverandre. Gudene vet hvor hun får energien fra. Hun er veldig flink til å bygge broer og se sammenhenger som jeg tror andre ikke helt har kapasitet til.»

– Teammedlem

Prosjektets mål, fremdrift og milepæler ønskes tydeligere kommunisert og enklere å finne frem til av en saksbehandler. Det påpekes at det kanskje ikke er vedkommende sin oppgave å tenke på dette, men at siden vedkommende faktisk lurer på det, så burde det ha vært en bedre ansvarsavklaring i starten. Med en tydelig ansvarsavklaring tror vedkommende at saksbehandlere vil bruke mindre tid på irrelevante ting. Man kan finne ut av hva som forventes av hverandre og hva som forventes av en selv.

Et utviklingsteammedlem påpeker at det er stort arbeidspress både hos saksbehandlere og hos utviklingsteamet. Det gjør det vanskelig å velge hvilke aktiviteter som skal prioriteres av teamet til videre utvikling. Man må gjøre ting som gir verdi for alle fordi man er så presset på tid. På grunn av covid-19 er det flere som er syke og som dermed søker om sykepenger.

Sykefraværet hos saksbehandlerne er like stort som ellers i landet, så det blir flere saker og færre som kan behandle disse sakene.

I dag blir saksbehandling målt på hvor mange saker man klarer å saksbehandle, ifølge et teammedlem. Vedkommende mener det blir feil målegrunnlag fordi sakene ikke er like. Å måle på antall saker sier ikke noe om kvaliteten på behandlingen av sakene. Vedkommende påpeker at det er for få som jobber med behandling av sykepenger og at saksbehandlerne trenger mer tid til hver sak.

Både teammedlemmer og saksbehandlere anser det som positivt at det er færre brukere av systemet til daglig. Det gir brukerne mer tyngde da de bruker systemet mer og er bedre kjent med hva som er systemets vanlige oppførsel. Brukerne vil dermed oppdage unormal oppførsel tidligere og si ifra om det, noe som både saksbehandlere og teammedlemmer trekker frem som elementer som bidrar til enklere samarbeid.

Konstruerte brukertester anses av teammedlemmer som oppkonstruerte og at de er preget av egne bias. Verden er ofte mye mer kompleks, så man må ha gode metrikker og monitoreringsverktøy for å fange opp feil, heller enn å bruke mye tid på å teste seg til trygghet på forhånd.

Det er observert at et teammedlem opplever at det er for lite forståelse i organisasjonen for viktigheten av brukermedvirkning i riggen av sykepenger. Det blir brukt mye tid og energi på å overbevise om at brukermedvirkningsaktivitetene, spesielt saksbehandlertreff, er viktige.

Teammedlemmer sier at det kan oppstå utfordringer der man må utenfor teamet for å finne løsninger. Da må man finne ut hvordan den informasjonen skal flyte og hvordan man skal ta tak i det som dukker opp.

En saksbehandler som har vært gjennom prosjektgjennomføring der fossefallsmetoden ble benyttet, ser virkelig nytten fra et brukerperspektiv ved å benytte smidig metode. Ved prosjektet som ble gjennomført med fossefallsmetoden fremhever vedkommende at det ble levert et ferdig produkt der det tok veldig lang tid å få innført nødvendige endringer. Vedkommende setter pris på at man ved smidig produktutvikling kan gjennomføre nødvendige endringer mye raskere, noe som gir en bedre brukeropplevelse tidligere.

5.2 Holdninger og motivasjon til brukermedvirkningsaktiviteter

Teammedlemmer sier de har inntrykk av at det, innad i Team Bømlø, er god forståelse for at brukermedvirkning er viktig i utviklingsprosessen. Ellers i organisasjonen har de et inntrykk av at det er varierende holdninger og motivasjon til brukermedvirkning, men at de begynner også å få øynene opp for metoden og anerkjenne det som en riktig metode for å lage gode løsninger. Videre sier teammedlemmer at alle i utgangspunktet virker positive til brukermedvirkningsaktiviteter, men at det varierer litt etter personlighet, erfaringer og arbeidsoppgaver man har. Hvor mye man faktisk gjennomfører slike aktiviteter avhenger av hvordan hver enkelt prioriterer tiden sin. Et eksempel er saksbehandlertreffene (se kapittel 5.6) som er frivillige å delta på. Alle, både saksbehandlere og utviklingsteammedlemmer, får invitasjon til å delta på treffene, og designerne fordeler treffene slik at det blir jevnt fordelt.

Gjennom observasjon virker det som at brukerne blir hensyntatt i stor grad. Det er observert et tilfelle i Slack der utviklingsteamet har tatt med et problem fra brukerne videre opp i organisasjonen for å få det løst slik brukerne mener er mest hensiktsmessig.

Klikkbare prototyper benyttes der designerne mener at det kan gi bedre beslutninger og forståelse av hva de har tenkt. De har også begrenset bruk av konstruert brukertesting. Det er *begrenset* bruk av dette fordi det anses som vanskelig å finne ut om man får testet de riktige tingene i en slik brukertest.

Det observeres at brukernes innspill blir tatt hensyn til gjennom arbeidsdagen til utviklingsteammedlemmene i høy grad. Det kan stilles spørsmål til viktigheten av et problem hvis det kun er én bruker som etterspør noe. Da undersøkes det om det er flere brukere som mener det samme, eller om vedkommende faktisk er alene om å mene eller ønske seg dette. Permanente løsninger som faktisk skal være med i den fremtidige løsningen prioriteres over å løse midlertidige problemer.

Det er observert at det blir nevnt universell utforming i beslutninger som tas. Et observert eksempel er at fargeblinde også skal kunne tyde elementer som fremstilles gjennom fargekoder i systemet.

5.3 Kommunikasjon

Språket innad i utviklingsteamet observeres som lett, saklig og det benyttes en hyggelig tone og humor. Det virker som at alle er enige om hva som kommuniseres. Det observeres at folk er positive og åpne til de forslagene som blir lagt på bordet. Det virker som at det er god stemning innad i teamet. Det observeres at det varierer hvor fort utviklingsteamet kommer frem til hva som er problemet. Det kommuniseres også frem og tilbake om hva som skal få høyest prioritet av for eksempel hvorvidt man skal vekte brukeropplevelsen tyngre enn kode/automatikk.

Både saksbehandlere og teammedlemmer sier at det skjer misforståelser mellom partene. Flertallet av intervjuobjektene sier at misforståelser blir løst på en god måte, og ingen av intervjuobjektene har sagt at de ikke blir løst eller at de løses på dårlige måter. Det er relativt få brukere av systemet, så det er mulig å kommunisere direkte med brukerne.

Hvis saksbehandlere lurer på hvorfor et bestemt element er løst slik det er løst rent teknisk, anser et teammedlem det som en utfordring å kommunisere dette på et passende teknisk nivå. Teammedlemmet løser dette ved å starte med helt enkle forklaringer og så gå mer og mer detaljert til verks. Tegninger er et godt hjelpemiddel som også benyttes.

Et teammedlem påpeker at hvis man ser på samhandlingen mellom utviklingsteamet og saksbehandlerne som at det er fagpersoner som diskuterer en problemstilling istedenfor å kalle det brukermedvirkning, kan det gi en bedre arbeidsdynamikk. Hvis man lykkes med slike diskusjoner, mener vedkommende at det virkelig gir fremdrift. Vedkommende påpeker at det også er veldig nyttig å møtes ansikt til ansikt og prate om problematikken. Verktøyene er å forstå problemet på lik måte, at alle har samme fokus og at man greier å møtes og ha kjennskap til hverandre på et personlig plan.

En saksbehandler påpeker at kommunikasjonen utenom saksbehandlertreffene må være god. De må vite hvor de skal henvende seg, og de må ha en ramme for hvordan spørsmålene kommer frem.

Det anses som en utfordring fra noen av saksbehandlerne som ble intervjuet at utviklingsteamet ikke forstår hvorfor noe er viktig for dem. En saksbehandler sier det er forståelse for at det er komplekse tankeprosesser som ligger bak saksbehandlingsrutiner og prosessene som settes i gang for å behandle en sak. Likevel opplever de at det oppstår problemer ved kommunikasjonen rundt dette, men at det løser seg ved at man er tålmodig med hverandre, tilføyer mer informasjon underveis og er åpne for å motta oppfølgingsspørsmål, ifølge en saksbehandler. Flere fra utviklingsteamet anser det som en utfordring å kommunisere godt nok til saksbehandlere at etterspurt funksjonalitet ikke er på toppen av prioriteringslista akkurat nå. Det kan være faglige uenigheter om hva som er best for produktet. Utviklingsteamet må se funksjonalitetene til produktet i et større perspektiv, både i forhold til fagområdene som er representert, men også i form av andel av brukerne som mener det er et behov. Utviklingsteamet jobber stadig med å lytte og veie opp hva som kan gjøres først og om det gir mening å gjøre det på den foreslåtte måten. Brukerne blir lyttet til, men utviklingsteamet må også bruke sin kunnskap til å gi produktet den mest egnede funksjonaliteten. Et annet element er å lytte til hva de prøver å formidle. Et eksempel er hvis en bruker sier at «denne må være rød», men det de ønsker er at «denne» skal være mer synlig.

En saksbehandler påpeker at det å være med på utviklingsprosessen og å jobbe så tett med utviklingsteamet gir saksbehandlerne bedre forståelse for hvorfor ting løses slik de løses og hvorfor ting tar den tiden det tar. Vedkommende føler at det tette samarbeidet gir begge parter bedre forståelse for hvordan den andre parten jobber. Dette hjelper utviklingsteamet med å kunne utvikle det mest hensiktsmessige produktet. Flere saksbehandlere påpeker at de synes utviklingsteamet er flinke til å forstå deres behov og arbeidsprosesser.

«Jeg har veldig trua på å være såpass nært de som utvikler løsninger, men samtidig at de er så nært oss slik at de får en helt annen forståelse for hva vi trenger i praksis.»

– Saksbehandler

Et mål med det nye saksbehandlingssystemet er at det skal effektivisere og redusere antall saksbehandlere det er behov for gjennom å automatisere behandlingen av mange saker. Automatiseringen har ikke blitt gjort i så stor grad som saksbehandlerne hadde blitt lovet, noe som har ført til at forventninger har blitt brutt. Ifølge et teammedlem har utviklingsteamet fått høre at de ikke har levert godt nok, noe de opplever som tungt å høre når man gjør så godt man kan. Dette anses av teammedlemmet som det vanskeligste med hele produktet. En fra utviklingsteamet påpeker at kontinuerlig dialog om planer som teamet har for utvikling, hva det er de jobber med og hva de tenker skal prioriteres fremover må bli bedre. Med en gang det er brukergrenseflater, brukergrensesnitt og flyt, så er det nyttig å ha brukerne med

i dialogen, både tidlig i utviklingen og underveis. Forslag til gjennomføringsmetode for dette er gjennom tett samarbeid hele veien og god dialog.

Et forslag til en brukermedvirkningsaktivitet fra et utviklingsteammedlem er å implementere et finkornet tilbakemeldingsverktøy i løsningen. Et eksisterende eksempel på dette som trekkes fram er tilbakemeldingsknapper som ofte finnes på nettsider i dag der man blir spurt om man fant det man lette etter. Tanken er at teamet skal kunne få tilbakemeldinger på konkrete elementer i selve saksbehandlingssystemet. Vedkommende mener man da kanskje kan slippe å sette opp møter med fire til fem personer for å finne ut av disse tingene. Et forslag til løsningsmetode fra vedkommende er at saksbehandlerne kan markere det elementet i saksbehandlingsbildet som ikke fungerer optimalt og sende det inn. Det vil da sorteres og kanskje også legges inn i Trello. Ved å innføre denne typen tilbakemeldinger er håpet til teammedlemmet at man bruker minst mulig tid på kommunikasjonsbarrierer og mer tid på selve problemet. Det er god forståelse for at denne løsningen kan være svært tidkrevende å få til utviklingsmessig slik at det muligens ikke er hensiktsmessig.

Det er allerede faste digitale møter på en halvtime i uka der utviklingsteamet fokuserer på hva de har gjort den siste tiden og deler hva de fokuserer på fremover. En fra utviklingsteamet frykter at det er for mange i møtet til at det blir noe særlig dialog og om det gir verdi begge veier. Det er dog en fin informasjonskanal. De prøver også å ha en åpen dør til all mulig kommunikasjon. Likevel tror vedkommende det er lurt å ha noen faste møter som utgangspunkt, for det å si at man skal ha god kommunikasjon og ha åpen dør der man kan spørre hvis det er noe, fungerer ikke alltid alene.

Kommunikasjonen fra utviklingsteamet anses av en saksbehandler å være «litt av og på». Som ny inn i prosjektet opplevdes roller i teamet som diffuse, at det var lite oversikt og at ting var uklart på generell basis. Etter hvert som vedkommende har kommet inn i prosjektet, kommer folk tydeligere frem i Slack, i møter og i andre informasjonsskriv. Vedkommende føler at folk sier ifra om ting på en generell basis.

En saksbehandler skjønner at det er mye arbeid som ligger bak innføringen av en ny saksbehandlingsløsning. Likevel mener vedkommende at formidlingen av det enorme arbeidet kunne vært bedre ut til saksbehandlerne som kun jobber i *gammel* løsning.

Kommunikasjon gjennom digital skriftlig kommunikasjonskanal, Slack

Kommunikasjonsplattformen Slack benyttes aktivt og utviklingsteamet svarer hyppig på henvendelser der. Utviklingsteamet mener det er lav terskel for å legge ut spørsmål og kommentarer på Slack. Noen saksbehandlere synes derimot det har vært høy terskel for å legge ut spørsmål og bemerkninger til løsningen på Slack. Dette har vært i frykt for at det de lurer på er dumt, om noen allerede har spurt om det de lurer på eller om de burde skjønt svaret selv. Det tar litt tid å bli trygg på plattformen da det oppleves av noen å være en helt annen måte å snakke på og å diskutere på enn å ta det ansikt til ansikt. Det har også blitt nevnt av saksbehandlere at taggesystemet er avansert og at det ønskes en bedre innføring i dette.

Språkbruken mellom partene i Slack observeres til å oftest være saklig, med noe humor innimellom. Det er observert noen tilfeller der det virker som det er frustrasjon hos brukerne

ut ifra hva som skrives på Slack. Totalt sett virker det som at brukerne forstår hva utviklingsteammedlemmene ønsker å formidle via Slack.

Utviklingsteamet observeres å være flinke til å informere via Slack hvis det skal foregå arbeid på systemet som kan medføre at systemet er nede. Saksbehandlerne informeres om hvorfor, og ofte hvor lenge, systemet er nede. Det er likevel observert et tilfelle der utviklingsteamet informerer om dette i etterkant der de i tillegg sier ifra om at de burde ha informert tidligere om nedstengingen av systemet. Tilbakemeldingene i Slack viser at folk var forståelsesfulle for dette. Utviklingsteamet virker også flinke til å informere saksbehandlerne på Slack om hva som viste seg å være årsaken til at et problem har oppstått.

Det er observert at ved lansering av ny funksjonalitet medfølger dokumentasjon via Slack om når funksjonaliteten skal benyttes og når den ikke skal benyttes, blant annet gjennom steg-for-steg-beskrivelser av dette. Dersom det observeres at saksbehandlere bruker systemet på en annen måte enn hvordan det er tenkt, eller at det observeres feil praksis, kan tekstene og de nye lanseringene benyttes til å motarbeide dette. Designerne må, ofte sammen med jurister, finne ut av hvordan det kan legges til rette for riktig atferd i systemet og hva som er viktig å informere om ved nye lanseringer. Det observeres at lanseringer skjer på en ryddig måte.

Det er observert at både utviklingsteammedlemmer og saksbehandlere er gode til å kommunisere problemer med få og tydelige meldinger frem og tilbake via Slack. Det er allikevel observert et tilfelle der det virker som at utviklingsteamet ikke helt forstår viktigheten av situasjonen på samme måte som saksbehandlerne. Det går mye kommunikasjon frem og tilbake i denne tråden på Slack, der begge parter forsøker å gi den andre parten en bedre forståelse av sin egen situasjon. Til slutt kommer de frem til en løsning der utviklingsteamet «håper» de skal få avklart problematikken slik at den typen situasjoner ikke vil oppstå i senere tid. Det er ikke observert noe mer kommunikasjon rundt problemet i ettertid.

Det har blitt observert to tilfeller der det virker som at saksbehandler ikke helt har forstått hva som formidles fra teammedlemmer via Slack. I det ene tilfellet ble det dermed benyttet videochat for å enes om problemstillingen. Det har ikke kommet frem om, og i så fall hvordan, det andre tilfellet ble løst.

Det er relativt hurtig responstid fra utviklingsteamet på Slack; fra problemet er sendt i Slack til utviklingsteamet sier ifra om at de skal ta tak i det, eller til de informerer om at saken tas tak i når tiden strekker til. Ikke alle saker blir besvart med en gang, men de blir besvart hensiktsmessig hurtig. Utviklingsteammedlemmer er flinke til å tagge folk de ser kan besvare det gjeldende spørsmålet eller saken. Det er observert én sak fra saksbehandler som ikke har blitt tatt tak i eller kommentert av noen fra utviklingsteamet. Grunnen til dette er ukjent.

Det er observert et fåtall tråder i Slack der sakene virker å være uoppklart. I et fåtall tilfeller er det observert at utviklingsteammedlemmer sier at de skal komme tilbake til saksbehandler med mer informasjon, uten at det har blitt gjort. Det observeres en holdning fra utviklingsteamet at hvis noe ikke blir tatt tak i av teamet med en gang, og det haster fra saksbehandlerne sin side, så vil det bli tatt opp på nytt av saksbehandlerne. Likevel virker det som at de fleste saker blir oppklart.

Gjennom observasjon i Slack virker det som at utviklingsteammedlemmene skjønner brukernes situasjon i stor grad. De kommer med forslag til permanente og ikke-permanente implementeringer i systemet som skal gjøre det lettere for brukerne.

5.4 Dokumentasjon

En saksbehandler synes begge parter er gode på kommunikasjonsprosesser fortløpende. Likevel ønsker vedkommende at dokumentasjonen skal være bedre slik at man ikke mister informasjon og at man slipper å gjenta ting som blir gjort eller sagt underveis. Dette er for at hvis man har gått glipp av noe, så kan man lese seg opp på hva som er gjeldende akkurat nå. Dette gjelder spesifikt nå når det skjer endringer hele tiden. Da må man vite hva som gjelder nå og hva som allerede er utdatert. Kan man forvente at det fortsatt oppstår samme feil i systemet, eller er det fikset? Saksbehandleren påpeker at informasjonen som deles gjennom lapper på Trello er en type oversikt, men det er skrevet med ord og uttrykk som er ukjent for saksbehandlerne, og som vedkommende føler heller er ment for utviklerne. Det er likevel forståelse for at det er vanskelig å holde dokumentasjon oppdatert. En annen saksbehandler ønsker også mer dokumentasjon og oversikt over hvilke prosesser som foregår til enhver tid fra utviklingssiden og fra produkteiersiden. Vedkommende synes det er mangel på en god plattform for en slik oversikt slik det er i dag.

«Personlig tror jeg at det hadde vært nyttig med litt mer dokumentasjon og oversikt over hvilke prosesser som foregår til enhver tid fra utviklingssiden og fra produkteiersiden. Jeg tror vi hadde hatt nytte av å ha tilgjengelig dokumentasjon over hva som foregår. Vi mangler en god plattform for en slik oversikt.»

– Saksbehandler

Ifølge teammedlemmer har koden til systemet versjonshåndtering. Det vil si at når man endrer i koden, så kan man fortsatt gå tilbake og se hva som var gjeldende kode på et tidligere tidspunkt. Hvilke paragrafer som benyttes til de ulike funksjonalitetene legges inn i den gjeldende koden slik at det alltid vil være tilgjengelig der. Jurister kan dermed, i samarbeid med utviklere, gå inn i koden og verifisere at funksjonaliteten følger lovverket. Juristene kan også teste seg frem til å verifisere dette. Det kan i så fall gjennomføres ved å opprette testpersoner som slår ut på forskjellige caser for å se om systemet viser riktig i løsningen. Lovverket benyttes som et oppslagsverk.

Det har blitt oppdaget en feil i en ordlyd i det nye systemet som, hvis ikke oppdaget, ville ført til feil i utbetalingen til søker. Det hadde blitt brukt mye tid på denne ordlyden for at den skulle bli god og riktig.

5.5 Supersaksbehandlerrollen

Ifølge et teammedlem skal supersaksbehandlergruppa bestå av folk fra flere enheter i NAV fordi «man gjør ting litt forskjellig fra enhet til enhet». Vedkommende mener at det ikke er god nok representasjon fra de enhetene som er mest ulike. Derfor ønskes det å bygge ut denne gruppa på sikt slik at det blir bedre representasjon. Det er ikke dokumentert hvordan

supersaksbehandlerne ble valgt ut til å begynne med, ut over at de skal representere forskjellige enheter i NAV.

En saksbehandler sier at vedkommende er usikker på hva som begrenser supersaksbehandlerne sitt ansvar. Dette mener vedkommende kunne ha blitt avklart tidligere i opplæringen til saksbehandlerne.

En saksbehandler uttrykker frustrasjon over at alle spørsmål vedrørende utvikling av det nye systemet skal kanaliseres gjennom supersaksbehandlerne. Samtidig er det forståelse for at hvis alle skal spørre utviklingsteamet om alt mulig, blir det tungvint og unødvendig for teamet. Saksbehandleren føler at det er supersaksbehandleren som vurderer hvor viktig spørsmålet er. Vedkommende har fått til svar at «nei, det er ikke vits å ta det nå», eller at «det har vi prøvd før», noe som oppleves frustrerende når det føles viktig for vedkommende selv. Saksbehandleren opplever det som at ting stopper opp hos supersaksbehandlerne. Saksbehandleren føler at hvis man har jobbet en stund og bygget opp en «status», kan man bli bedt om å spørre om det man lurer på i Slack.

Utviklingsteammedlemmer påpeker at når man har supersaksbehandlere som man blir bedre kjent med enn «vanlige» saksbehandlere, gir det en annen dialog med disse brukerne. Begge typene dialog anses som positivt fordi det gir forskjellige elementer til utviklingen. Saksbehandlere som man har snakket mye med skjønner ofte mer av utviklingsprosessen. Det er bra, men det kan føre til at de velger å ikke dele problemer eller utfordringer fordi de tenker at «teamet løser det snart», noe som ikke trenger å være tilfellet. Teammedlemmene opplever at saksbehandlerne de ikke kjenner så godt deler alt de sliter med og ting som ikke fungerer.

5.6 Saksbehandlertreff

Den største brukermedvirkningsaktiviteten teamet gjennomfører er observasjon av saksbehandlerne i arbeid, som kalles saksbehandlertreff. Intervjuobjektene har vært med på alt fra to til ti saksbehandlertreff. Treffene gjennomføres digitalt hver uke i regi av designerne. Det gjennomføres ved at en eller to fra teamet, uavhengig av rolle, møter en saksbehandler i ca. 30 min og observerer hvordan saksbehandleren jobber i systemet. Et teammedlem presiserer at de ikke er knallharde på *observasjon* som metode og at det derfor heter saksbehandlertreff. Ordet «observasjon» benyttes likevel om denne aktiviteten i denne oppgaven. Teamet benytter treffene til å lære hvordan saksbehandlerne faktisk benytter systemet. Fra denne kunnskapen kan de styre saksbehandlerne sin atferd i systemet i riktig retning i forhold til jussen. Teamet ønsker at saksbehandlerne skal være trygge på teammedlemmene som observerer dem i arbeid. Utviklingsteamet kan ønske å observere behandling av spesifikke saker eller av tilfeldige saker samtidig som saksbehandler forteller høyt hva som gjøres og hvorfor. Saksbehandler kan stille spørsmål til utviklingen, eller til spesifikke elementer med systemet under treffet. De kan også vise hva de trenger av løsningen, noe som flere saksbehandlere har påpekt som veldig nyttig. Teammedlemmene stiller også spørsmål for å få en bedre forståelse, men kommer ikke med korrigerende forslag. Dette er for å få det mest reelle bildet av saksbehandlers forståelse og bruk av systemet og å innhente så rene og upåvirket data som mulig. Teammedlemmer kan stille spørsmål for å undersøke og innhente informasjon om de forskjellige problemstillingene, og skal i den prosessen ikke påvirke selve saksbehandlingen. Gjennomføringen av treffene avhenger derfor

enten av hva saksbehandler ønsker eller lurer på, eller hva utviklingsteamet trenger av det gjeldende treffet. Treffene oppleves av alle intervjuobjektene som å gi stort verdibidrag til utviklingen, blant annet fordi det er så nakent. Treffene benyttes også til å danne en relasjon mellom saksbehandlere og teammedlemmene. Dette er for å gjøre det tryggere for saksbehandlerne å si ifra om hva enn det skulle være ved en senere anledning. Det påpekes også av saksbehandlere at det er enklere å ta opp ting i mindre grupper, i motsetning til hvis man legger ut en melding på Slack.

Alle som har blitt intervjuet synes at saksbehandlertreffene er nyttige. Dette begrunnes med at det gir enorm verdi å se hvordan produktet *faktisk* blir brukt. Produkter oppleves av intervjuobjekter å ofte bli brukt forskjellig i forhold til hva som er tenkt. Dermed kommer det frem feil og mangler som man kanskje ikke har tenkt på tidligere.

«Det er nesten helt innlysende at det å se hvordan produktet blir brukt gir masse verdi. Det (produktet) blir nesten aldri brukt slik du hadde tenkt. [...] Man får et fjes å knytte til problemene (i produktet). Da får man en helt annen indre motivasjon og en helt annen kontekst å putte problemene i.»

– Teammedlem

Det benyttes også uformelle treff med saksbehandlere. Et eksempel på dette er hvis utviklingsteamet har lyst til å observere systemet sin oppførsel for en spesifikk type sak, så kan de be saksbehandlere ta kontakt hvis de kommer over dette. Da blir saksbehandleren invitert inn i et Zoom-rom der de kan diskutere løsningen sin oppførsel i forhold til den typen saker. Dette kan minne om formelle saksbehandlertreff, men er mindre formelt. I resten av oppgaven omtales formelle saksbehandlertreff som «saksbehandlertreff» eller «treff». Hvis det er snakk om uformelle treff, vil dette være spesifisert. Fysiske uformelle samtaler med saksbehandlere på kontoret, for eksempel der man diskuterer tregheter i systemet ved kaffemaskinen, påpekes også som nyttig av et utviklingsteammedlem. Tregheter i systemet er noe som ikke nødvendigvis anses som så viktig at saksbehandleren ville sendt en melding i Slack, men anses av teammedlemmet som viktig å få frem likevel.

Både formelle og uformelle saksbehandlertreff anses som ekstremt nyttig, spesielt siden det utvikles i et syntetisk miljø der dataene er *litt* annerledes enn den er i reelle sykepengesaker. Ifølge en saksbehandler er de mest vanlige tekniske feilene som dukker opp feil som er kjent fra før, eller at det er gamle feil som kommer tilbake igjen. Typen feil som blir oppdaget er tekniske feil, faglige «hull» i systemet eller brukeropplevelsesfeil. Brukeropplevelsesfeil vil si for eksempel at elementer burde flyttes eller at det er for mange museklikk for å komme til en funksjonalitet. Det oppdages også hvordan og hvorfor saksbehandlere gjør ting litt ulikt gjennom disse treffene. Et mål i utviklingssammenheng er å utvikle en løsning som holder seg helt og holdent innenfor der alt er juridisk korrekt. Da gir det verdi å finne ut av hvor saksbehandlere flest befinner seg. Deretter kan man justere systemet slik at alle saksbehandlerne holder seg helt og holdent innenfor lovverket. Saker skal vurderes likt uavhengig av hvilken saksbehandler man får. Hvis utviklingsteammedlemmer oppdager rent

juridiske feil under et saksbehandlertreff, tar man det ikke opp der og da. I etterkant kan teamet gi tilbakemelding til alle saksbehandlerne om for eksempel feil praksis som har blitt observert gjennom treffene eller ting som er viktig å huske på i de forskjellige situasjonene. Det anses som lite produktivt i lengden å ta opp denne typen ting under selve møtet, da det kan føre til at saksbehandlerne blir mindre trygge på å dele sine tanker i frykt for at det skal være feil. Saksbehandlere opplever at feilene som oppdages og tilbakemeldingene deres blir tatt seriøst og tatt med videre i utviklingen. Dette kan bidra til tillitsbygging mellom partene.

«Det (brukermedvirkningsprosessen) er en viktig del av det å jobbe smidig. Det er jo stadig tilpasninger til behov, og våre kollegaer på design og utviklersiden ser og forstår nye problemstillinger underveis. Så vi kunne ikke ha jobbet på noen annen måte enn å ha sånne treff. Hvis de skal få en innsikt i vår hverdag, alt fra hvor vi kikker på skjermen til hva som er prosessen bak den konklusjonen vi har kommet frem til, så har de behov for å ha disse møtene.»

– Saksbehandler

Flere saksbehandlere føler det er viktig å benytte denne sjansen til å vise og forklare hva de trenger av løsningen. Hva trenger de mer av, og hva trenger de mindre av. De får også en bedre forståelse om hvorfor løsningen er som den er gjennom treffene, ved å snakke direkte med utviklingsteammedlemmer. Saksbehandlere anser det som fint og godt å få den tiden på treffet til å kunne si ifra om ting og å kunne jobbe i løsningen individuelt. Saksbehandlerne er forskjellige og legger vekt på forskjellige ting i løsningen, så det oppleves som godt å få delt sine tanker så direkte.

Saksbehandlere opplever kommunikasjonen fra utviklingsteammedlemmene som god under treffene. Noen saksbehandlere har poengtert at begreper kan være forskjellige fra de forskjellige partene, men at hvis noe er uklart er begge parter tydelige på det.

To fra utviklingsteamet sier at den største utfordringen ved treffene er at saksbehandlere kan føle seg «testet», noe som ikke er funksjonen til treffet. Teammedlemmer anser det som ekstremt viktig å presisere at treffene ikke er en test på faglig kvalitet hos saksbehandler, men at det utelukkende er for læring. For å bli tryggere på prosessen rundt treffene anbefaler flere intervjuobjekter, både saksbehandlere og teammedlemmer, at man er med på flere enn ett treff. Det går også på tillit og at man må ha en kultur der ingen spørsmål er for dumme. Dermed får forhåpentligvis begge parter mer ut av aktiviteten.

Flere saksbehandlere er positivt overrasket over hvor flinke utviklingsteammedlemmer er til å sette seg inn i deres situasjon. Saksbehandlere føler også selv at de klarer å se hva slags utgangspunkt utviklingsteammedlemmene kommer fra og justere sin kommunikasjon ut ifra det. Utviklingsteammedlemmer opplever at saksbehandlerne er flinke til å dele hva de gjør og hvorfor de gjør det. De opplever saksbehandlerne som flinke og engasjerte. Hvis de ikke hadde hatt et engasjement rundt jobben sin, tror utviklingsteammedlemmer at ingen av partene hadde fått like mye ut av saksbehandlertreffene. De opplever det som at det er mange

som bryr seg om å gjøre jobben sin på en bedre måte, og dette anses som viktig for at treffene skal fungere som tenkt.

Utviklingsteammedlemmer påpeker at det er viktig at utviklingsteamet får føle på kroppen hvordan systemet virker for de som saksbehandler i det. Da får man en helt annen indre motivasjon og helt annen kontekst å putte problemene i. En ting er å vite at en feil «kun» oppstår i 1 av 10 tilfeller, men når man da får se hvor vondt det gjør ved det ene tilfellet, så gir den en helt annen følelsesmessig tilknytning til problemet.

Noen saksbehandlere har poengtert at treffene hjelper dem med å skjønne det nye systemet bedre. De opplever det som at når de får en bedre forståelse for hvordan systemet er bygget opp, er det lettere å stole på at systemet tar riktige avgjørelser. Dermed føler de ikke et behov for å dobbeltsjekke ting som systemet allerede har sjekket.

Utviklingsteammedlemmer nevner at man blir tvunget til å se ting i et større perspektiv ved å delta på treffene og de får mer glede av å utvikle systemet. Et teammedlem belyser saksbehandlertreffene som nyttig for alle i alle faser. De som har vært i teamet lenge kan benytte det som en oppfriskning på hvordan systemet faktisk brukes. Nye personer i teamet kan benytte det som en intro til hva produktet er og hvordan det brukes. Flere teammedlemmer påpeker at man fort kan miste perspektiv og kontakt med brukerne når mye haster og det er mye som skal gjøres, så da er det godt å ha slike treff der man kan få denne innsikten.

5.7 Finansiering av utviklingen

En annen problemstilling som et utviklingsteammedlem trekker frem, er at NAV bør finansiere prosjektporteføljen sin på en annen måte. Vedkommende mener at prosjektfinansiering ikke passer til smidig arbeidsmetode. Likevel mener teammedlemmet at smidig produktutvikling anses å være den beste metoden for IT-produkter. Videre sier vedkommende at det i dag blir satt av for lite penger til å vedlikeholde produkter, så produktene blir for gamle og må så oppdateres, men så har man ikke nok penger til det. For å få finansiering sier vedkommende at man må søke om særskilte midler fra statsbudsjettet. Teammedlemmet ønsker en annen ærlighet i forhold til hva det koster å ha det regelverket man har i dag. Enten må regelverket implementeres i kode og i saksbehandling, noe som koster. Ellers må man forenkle regelverket, og det koster også. Sistnevnte påpekes også av en saksbehandler. Det koster masse i juridisk tid, men også mye i politikk, fordi regelverket er komplisert. Et eksempel som tas frem er jordbruk og reindrift som har et eget regelverk. Teammedlemmet mener det er store problemstillinger som krever tøffe beslutninger på høyt nivå, som er vanskelig for dette utviklingsteamet å løse.

6 Diskusjon

I dette kapitlet diskuteres temaer fra kapittel 4 og 5 opp mot teorien som er presentert i kapittel 3 eller opp mot andre funn. Temaene som diskuteres er de som anses relevant i henhold til brukermedvirkningsprosessen og kobles sammen med elementer eller teori som bygger opp under dette.

6.1 Organisering

To av hovedutfordringene i prosjekter er involvering av kunden og overgangen til at brukerne begynner å bruke produktet (Mohagheghi & Jørgensen, 2017, p. 371). Team Bømlo jobber smidig og produktbasert, og har dermed kontinuerlige leveranser til brukerne sine uten sluttdato. De får hele tiden tilbakemeldinger på ting som må forbedres og hva som fungerer fint. Kunden, eller brukeren, er dermed hele tiden involvert i utviklingen. De som gir tilbakemeldinger er brukere av systemet, så «overgangen til at brukerne begynner å bruke produktet» er gradvis. Når overgangen er gradvis, kan brukerne få mulighet til å bli bedre kjent med systemet og lære seg det samtidig som det utvikles. Når det er en kontinuerlig utvikling av produktet som ikke utvikles på prosjektbasis, vil det heller ikke bli en overlevering fra prosjektorganisasjon til kundeorganisasjon der kundeorganisasjonen må lære seg produktet for å kunne holde det vedlike. I dette tilfellet er organisasjonen som skal drive vedlikehold av produktet den samme organisasjon som har utviklet det, så de kjenner produktet ut og inn.

Det oppleves som stort arbeidspress både hos saksbehandlere og hos utviklingsteamet, noe som gjør det vanskelig å velge hvilke aktiviteter som skal prioriteres av teamet til videre utvikling. Saksbehandlere måles fra et høyere nivå i organisasjonen på antall saker som behandles, noe som kan medføre at de føler et enda større tidspress. Utviklingsteammedlemmer uttrykker at de føler seg presset fremover der man stadig må utvikle ny funksjonalitet. Både saksbehandlernes og utviklingsteammedlemmenes situasjoner kan tyde på at KPI-ene som de ansatte måles etter, ikke er helt ideelle. Ved å måle feil ting, kan det gjøre det vanskelig å skjønne hvilke beslutninger som må tas (Kerzner, 2017, p. ix) med tanke på ønsket arbeidsatferd og ønskede resultater. Det er viktig at de riktige KPI-ene velges slik at organisasjonen får frem den ønskelige arbeidsatferden. Det er uvisst om de ansatte måles av flere måleparametere enn de som er oppgitt her. Hvis saksbehandlerne for eksempel *kun* måles på antall saker de behandler, så kan man få behandlet mange saker nettopp fordi det er det man måles på. Dermed er det viktig at man måler de riktige tingene slik at ønsket arbeidsatferd kommer frem.

Teamets arbeidsform innebærer at det er stor aksept for at det kan skje feil og at man skal lære av disse feilene. Når man benytter smidige metoder er det anbefalt å holde brukerne kontinuerlig involvert (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32). En vinning fra denne kontinuerlige involveringen av brukerne i kombinasjon med smidig utvikling kan være muligheten til å kontinuerlig implementere den nye kunnskapen fra brukerne inn i produktet. Det er påpekt av flere teammedlemmer at teamet ikke har nok tid til å gå tilbake i allerede lansert funksjonalitet for å endre ut ifra lærdommen man har tilegnet seg. Et teammedlem sier at en grunn til dette er at de stadig blir presset til å levere ny funksjonalitet, noe som kan tyde på at teamets prestasjoner måles på feil grunnlag. Hensikten med å rapportere

prestasjoner er å vise hvilke aktiviteter eller handlinger som mottakeren av rapporteringen må overveie (Kerzner, 2017, p. ix). Hvis teamet ikke har tid eller mulighet til å benytte den nye lærdommen de har tilegnet seg, kan de miste denne viktige delen av kontinuerlig involvering av brukere i kombinasjon med smidig utvikling. Hvis teamet kun måles på fremgang, kan det føre til at kjent mangelfull funksjonalitet forblir mangelfull fordi teamet ikke har mulighet til å ta seg tid til å fikse det. En mulig løsning på dette kan være å måle andre prestasjoner når det kommer til fremgang av utviklingen; for eksempel gjennom måling av antall saker som blir behandlet *feil* gjennom den nye løsningen i forhold til lovverket, eller antall situasjoner der saksbehandlere ikke skjønner hva som menes i den nye løsningen. Begge disse målingene ønskes selvfølgelig å være så lave som mulig.

Fysiske uformelle samtaler med saksbehandlere på kontoret, for eksempel å diskutere tregheter i systemet ved kaffemaskinen, påpekes som nyttig av et utviklingsteammedlem. Når teamet (egentlig) er heldigitalt, kan det være viktig å få frem slike uformelle samtaler på annet vis slik at denne typen informasjon fortsatt kommer frem. Dermed kan det være nyttig å finne måter der teamet kan beholde denne verdien, også i et heldigitalt team.

En saksbehandler påpeker at noen av de mest vanlige tekniske feilene som dukker opp under et saksbehandlertreff er blant annet gamle feil som kommer tilbake igjen. Dette kan tyde på at det er manglende softwaretesting. Hvis det viser seg at feilen er kjent av utviklingsteamet, så er det ikke en «gammel» feil; det kan tyde på at det ikke kommuniseres godt nok mellom partene om når en feil eller bugg er helt løst. Et forslag til forbedring er at det ved all feilretting legges inn automatiske tester som feiler hvis den kjente feilen skulle gjenoppstå, og kommunikasjonen ut til brukerne om når en feil er rettet bør bli bedre.

6.2 Teamets arbeidsmetoder

Kommunikasjonen mellom utviklingsteammedlemmer gjennom arbeidsdagen skjer for det meste gjennom Slack og ved å poppe inn i forskjellige Zoom-grupperom. Dette kan minne om arbeidsdager i et kontorlokale. Når folk popper innom et grupperom der noen jobber, kan det stoppe arbeidsflyten til de som er i rommet fra før og det kan gi opphakkede dager. Dette kan også påvirke arbeidsflyten gjennom dagen da man må ta tak i de tingene som er viktigst *akkurat nå*. Dermed kan de planlagte arbeidsoppgavene for dagen måtte vike. Dersom dette blir et problem for teammedlemmer, kan det innføres en «terskel» for hvilke typer spørsmål som kan tas opp på denne måten gjennom arbeidsdagen. Et mulig tiltak for å unngå slike situasjoner kan være at hvis vedkommende som trenger hjelp vet at løsningen vil trenge mer enn ti minutter, så må de ta kontakt på annet vis enn å poppe innom.

Kvantitative data kan samles inn for eksempel gjennom spørreundersøkelser og monitorering av hvordan produktet brukes av brukerne; men man må nøye seg «med å måle ganske enkle forhold» (Jacobsen, 2021, p. 135) så det vil gi en begrenset forståelse av situasjonen. Teamet benytter statistikkpakker for å ha reelle data på hvordan funksjonalitet treffer brukerne. Dette kan dermed benyttes til å bestemme hva som bør endres, hva som bør legges til eller hva slags funksjonalitet som eventuelt bør fjernes. Likevel er det viktig å huske på at dette ikke nødvendigvis gjengir hele situasjonen, så det bør også benyttes kvalitative metoder.

For å kunne ta stilling til hvilke krav eller deler i standarden for universell utforming, WCAG, som av en eller annen grunn ikke er aktuelle for virksomheten, bør man ha kjennskap til

brukerne som skal bruke det ferdige produktet. Saksbehandlingssystemet som utvikles av Team Bømlo er komplekst og sammensatt. Systemet skal kun benyttes av et begrenset antall ansatte i NAV og teamet kan dermed finne ut av hvilke tilrettelegginger som kan være nødvendig for at alle saksbehandlere skal kunne bruke det med hensyn på universell utforming. Det er kun observert ett tilfelle der universell utforming blir nevnt. Observasjonsperioden er relativt kort, og det har kun blitt observert to arbeidsdager. Dermed kan det være et høyere fokus rundt dette enn det som er observert.

6.3 Kommunikasjon rundt felles mål, visjoner og arbeidsmetode

Flere av saksbehandlerne omtaler utviklingen av saksbehandlingssystemet som *et prosjekt*; en saksbehandler etterspør «prosjektets mål, fremdrift og milepæler». Saksbehandleren ønsker videre en tydeligere ansvarsavklaring i starten av utviklingen. Utviklingsteammedlemmer har spesifisert gjennom intervjuer og samtaler at utviklingen skjer på produktbasis og ikke på prosjektbasis og at roller og ansvar ligger mer på teamet som helhet heller enn på enkeltteammedlemmer. En saksbehandler og en fra utviklingsteamet savner begge en ansvarsavklaring i starten av prosessen. Ifølge et teammedlem har teamet med vilje vage roller slik at ikke rollen skal definere hva man kan bidra med og ikke. Teamets valg av å ha vage roller er i henhold til XP-prinsipper, der man har et mål om at alle skal bidra så godt de kan for teamets suksess (Beck & Andres, 2005, p. 82). At saksbehandlere omtaler utviklingen som et prosjekt kan være dagligtale, men det kan også tyde på uklarheter i forventninger om arbeidsmetode fra de forskjellige partene. Selv om teamet arbeider i henhold til teorien, så bør det vurderes om disse uklarhetene i forventninger om arbeidsmetode påvirker fremgangen i utviklingen og løsningene for produktet. Hvis dette er tilfellet, bør gjeldende arbeidsmetode klareres.

En fra utviklingsteamet påpeker at kontinuerlig dialog om planer som teamet har for utvikling, hva det er de jobber med og hva de tenker skal prioriteres fremover må bli bedre. En saksbehandler ønsker mer dokumentasjon og oversikt over hvilke prosesser som foregår til enhver tid fra utviklingssiden og fra produkteiersiden. Det er dermed et ønske fra begge parter om å bedre kommunikasjonen vedrørende fremdriften til utviklingen. Det benyttes mye kontinuerlig dialog i utviklingen allerede. Da det kommer frem som et savn fra begge parter, kan det tyde på at *måten* den kontinuerlige kommunikasjonen foregår på ikke er ideell. Ifølge Andersen (2018, p. 410) er det viktig hvordan brukerne blir involvert på. Dermed bør det utforskes hvordan denne kontinuerlige dialogen bør foregå hos utviklingsteamet Team Bømlo.

Et utviklingsteammedlem understreker at å diskutere problemstillinger som fagpersoner der man benytter hver sin fagkunnskap er en god arbeidsmetode som vil gi gode resultater. Dette er i tråd med Iivari og Molin-Juustila (2009, p. 55) som påpeker at brukere og utviklere bør anses som likeverdige parter der begge bidrar med sin spesifikke ekspertise. Dermed kan brukerne føle en mer tilhørighet til utviklingen, og følgelig bli mer engasjert. Dette kan videre føre til at teammedlemmene får flere og bedre innspill fra brukerne i utviklingsprosessen. Teammedlemmet sier ingenting om dette er noe som gjennomføres systematisk i dag, eller om dette kun er vedkommende sin fremgangsmåte. Likevel bør dette vurderes som et tiltak for å bedre den kontinuerlige kommunikasjonen med brukere.

Det viktigste ved softwareutvikling er verdien det gir til eierne eller brukerne (Robertson & Robertson, 2013, p. xxi). Noen av saksbehandlerne som ble intervjuet påpekte at en

utfordring er at utviklingsteamet ikke forstår hvorfor noe er viktig for dem. Flere fra utviklingsteamet anser det som en utfordring å kommunisere godt nok at etterspurt funksjonalitet fra saksbehandlerne ikke er på toppen av prioriteringslista akkurat nå. Brukere sier de har et behov for at utviklingsteamet skal forstå hvorfor noe er viktig; samtidig har et utviklingsteammedlem sagt at det er en utfordring å kommunisere til brukerne på et passende teknisk nivå hvorfor ting er løst på den måten det er gjort. Det kan stilles spørsmål ved om det er å snu brukermedvirkningsprosessen på hodet i og med at det er brukerne som skal dele sine behov til teamet og ikke motsatt. Hvis brukere føler at det er manglende forståelse fra teamet om viktigheten av spesifikk funksjonalitet, kan det stilles spørsmål ved om brukerne faktisk får kommunisert sine behov godt nok. Hvis brukerne mener at noe er vesentlig og burde prioriteres høyt, så kan det være gode grunner til det som bør komme frem ovenfor utviklingsteamet slik at de kan ta en velinformert vurdering. Saksbehandlere poengterer at treffene hjelper dem med å skjønne det nye systemet bedre. Når brukerne forstår systemet bedre kan det medføre at de kan bidra på en mer givende måte i utviklingen ved at forslagene er mer relevante enn hvis de ikke skjønner systemet. Det kan også gjøre dem mer trygge når de behandler saker i systemet og dermed ikke føle på en trengsel for å måtte dobbeltsjekke vurderinger som systemet allerede har sjekket, noe en saksbehandler har påpekt at kan være et tilfelle. På den andre siden kan forbedring av kommunikasjon fra utviklingsteamet til brukerne om teammedlemmenes faglige meninger og vurderinger gi brukerne bedre forståelse for de beslutningene som tas. Produktet er i stadig utvikling, så funksjonalitet som brukere anser som veldig viktig i dag kan være utdatert i morgen. Ved å kommunisere de større linjene i produktutviklingen til brukerne kan det gi brukerne bedre forståelse for det større perspektivet. Dermed kan utviklingsteamet være tryggere på at ønskene fra brukerne bør prioriteres så høyt som de sier at de bør, i og med at brukerne er velinformert om det større perspektivet. Utviklingsteamet bør også være åpne for at de har prioritert feil og at de faktisk burde prioritere den funksjonaliteten som brukerne etterspør akkurat nå. Dermed kan en god løsning være at brukerne blir bedre informert om de store linjene i utviklingen og at de får mer direkte innflytelse på prioriteringer.

Det er observert et tilfelle i Slack der det virker som at utviklingsteammedlemmer ikke er på samme bølgelengde som saksbehandlerne når det kommer til viktigheten av et problem. Utviklingsteamet «håper» de skal få avklart problematikken slik at den typen situasjoner ikke skal oppstå i senere tid. Ved å si at de *håper* kan dette oppfattes som en måte å ikke love bort noe, og dermed være en «snillere» måte og si at problemet ikke er topp prioritet akkurat nå. Det kan virke som at teammedlemmer ikke helt har tillit til brukernes vurdering av viktigheten av funksjonalitet. Dermed kan det også her være gunstig med god kommunikasjon om de store linjene slik at teamet er trygge på at brukerne skjønner dette; deretter kan en god løsning være at brukerne får mer innflytelse på prioriteringen. Det er ikke observert noe mer kommunikasjon rundt problemet i ettertid. Det kan ha blitt benyttet andre plattformer for videre kommunikasjon rundt problemet. Problemet kan også trenge lenger tid på å bli løst enn lengden på observasjonen i dette studiet, og dermed bli løst i senere tid. Dette funnet er et observert skriftlig tilfelle via Slack. Dermed kan kommunikasjonen oppleves annerledes for observator enn det som er ment fra avsender, og også enn hvordan det er tolket av mottaker.

6.4 Brukermedvirkning i startprosessen

Johansen et al. (2019, pp. 56–57) sier at det i begynnelsen av et prosjekt er «usikkerhet rundt fremtidige behov» med tanke på mål, leveranser og «fremtidige krav fra kunder og interessenter». Dette kan indikere at tidlig brukermedvirkning er ettertraktet for å finne ut av dette. Kilden gjelder spesifikt for prosjekter, men innholdet anses likevel som relevant i denne sammenheng. Ved å involvere brukerne i perioden der teamet finner ut av hva som faktisk trengs av produktet, kan man lære nettopp det. En saksbehandler savner bedre forarbeid ved å bli involvert i oppstartsprosessen. Utviklingsteammedlemmer har beskrevet deler av oppstartsprosessen der brukerne har blitt involvert gjennom diverse aktiviteter. Likevel kan det, gjennom savnet til saksbehandleren, virke som at de ikke føler seg inkludert i tilstrekkelig grad. Hva som anses som tilstrekkelig grad for denne saksbehandleren er uvisst. Ifølge Andersen (2018, p. 410) er det viktig hvordan brukerne blir involvert. Dermed kan det stilles spørsmål ved om måten brukerne ble involvert i startprosessen var ideell. Kanskje burde brukerne fått mer direkte innflytelse på prioriteringer. I denne sammenheng bør det også undersøkes hva slags savn det er; for eksempel om det er et savn vedrørende funksjonalitet som nå ikke er mulig å implementere grunnet valg som er tatt i oppstartsprosessen.

Beyer (2010, pp. 27–28) påpeker at utviklingsteam skal kartlegge nåværende situasjon og bli kjent med brukernes behov som videre hjelper dem med å forstå hvordan løsningen bør være som første del av utviklingen av et system. Utviklingsteamet i NAV gjennomførte denne kartleggingsperioden veldig likt som litteraturen tilsier. Første steg innenfor denne startprosessen er «å intervju brukerne på arbeidsplassen deres, se på dem i arbeid og snakke med dem om hva de gjør og hvorfor» (Beyer, 2010, p. 28). Dette gjennomførte utviklingsteamet ved å dra til arbeidsplassene til saksbehandlerne, observere dem behandle saker i det gamle systemet og å gjennomføre intervjuer med dem. Beyer (2010, pp. 27–40) sier at alle teammedlemmer kan trenes opp til å være med på kartleggingen, noe som sørger for at alle på teamet forstår brukernes problemer på et følelsesmessig plan. Utviklingsteamet i NAV gjennomførte denne fasen der forskjellige roller var med på observasjoner og intervjuer med saksbehandlere, akkurat som Beyer (2010, pp. 27–40) anbefaler. Selv om det er nevnt et ønske om bedre involvering i startprosessen av en saksbehandler, vurderes det til å ha vært tilstrekkelig i forhold til litteraturen. Det viktigste er likevel hvilken informasjon som har blitt funnet og benyttet av teamet til utviklingen av produktet.

6.5 Organisering av brukermedvirkningen

6.5.1 Generell brukermedvirkning

Flere teammedlemmer påpeker at det ene teammedlemmet er veldig flink til å involvere alle i teamet med hverandre, noe som anses som noe «andre ikke helt har kapasitet til». Designerne blir også påpekt av saksbehandlere som veldig flinke når det kommer til involvering av brukere. Det kan dermed stilles spørsmål ved om brukermedvirkningsaktivitetene til dette utviklingsteamet henger for mye på enkeltpersoner. Hva vil skje dersom disse personene blir byttet ut med andre som ikke har dette engasjementet for brukermedvirkning? Brukermedvirkning anses som en viktig del av det å jobbe smidig, både fra intervjuobjekter og fra litteraturen (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32; Yaman et al., 2016, p. 345). Dermed bør brukermedvirkningsaktiviteter

organiseres slik at de er en del av arbeidskulturen og ikke avhenger av spesifikke personer for å bli gjennomført, i noe større grad enn det som ser ut til å være tilfelle for Team Bømlø.

For lengre samarbeidsperioder med brukere veier det tyngst at brukerne kan bidra aktivt og gi direkte fremgang i utviklingen (Rasmussen et al., 2011, p. 186). Utviklingsteammedlemmer har sagt at de setter veldig pris på at saksbehandlerne er flinke til å kommunisere deres tanker. Flere tror også at hvis saksbehandlerne ikke hadde hatt det engasjementet de har rundt jobben sin, så hadde ikke saksbehandlerntreffene vært like nyttige. Dette kan vise at å «bidra aktivt og gi direkte fremgang til utviklingen» kan skje på forskjellige måter. Gjennom å ha et engasjement for jobben sin, kan man bidra til utviklingen. Man kan også bidra til utviklingen gjennom å klare å kommunisere tankene sine. Dette kan vise at det er engasjerte brukere som har blitt involvert i medvirkningsprosessen i denne casen. Likevel kan mindre engasjerte brukere også ha blitt involvert, men det har ikke blitt samlet inn funn vedrørende dette. På den ene siden er det forståelig å involvere engasjerte brukere som kan bidra til fremgangen av utviklingen. Likevel er det viktig å involvere et representativt utvalg av brukergruppa, slik at det ikke kun blir deler av brukergruppa sine behov som blir møtt.

Saeeda et al. (2020, p. 1) nevner at en hovedutfordring ved kravspesifikasjoner når man jobber med smidig softwareutvikling er å måtte inngå kompromisser på brukernes involvering som kan påvirke softwaresystemets funksjonalitet og kvalitet på en negativ måte. Beyer (2010, p. 18) gir et eksempel på dette som er at brukerne ikke alltid har tid til å delta i brukermedvirkningsaktiviteter. Det er ikke gjort funn fra intervjuobjektene som tilsier at brukere ikke har tid til å delta på brukermedvirkningsaktiviteter. Utviklingsteamet har en løsning der brukere inviteres til å delta på et saksbehandlerntreff, og de kan selv styre hvor aktive de ønsker å være på Slack og gjennom andre brukermedvirkningskanaler. Utviklingsteammedlemmer har også poengtert at saksbehandlerne oppleves som flinke og engasjerte. Dette kan medføre at de brukerne som deltar faktisk er engasjert og motivert for utviklingen av produktet, som kan anses som en god egenskap i utviklingen. Det kan dog diskuteres om disse brukerne som er frivillig aktive i brukermedvirkningsprosessen er den gjennomsnittlige brukeren av produktet. De kan absolutt være nyttige å ha med på laget, men man bør ha med andre typer brukere i tillegg. Det kan undersøkes om brukerne som ikke melder interesse til å delta på saksbehandlerntreffene, eller som er lite aktive i brukermedvirkningsaktiviteter, velger det bort fordi de ikke har tid eller om de velger det bort av andre grunner. Hvorfor saksbehandlere velger å delta mindre i brukermedvirkningsprosesser er ikke undersøkt i dette studiet.

De ukentlige møtene med saksbehandlere anses av et teammedlem å være en fin kommunikasjonskanal ut til brukerne. Brukerne har også etterspurt bedre innføring i fremgang i utviklingen. Fremgang, mål og ny funksjonalitet kan formidles bedre gjennom disse møtene. Referater, eller annen type dokumentasjon, av det som har blitt gjennomgått på disse møtene kan benyttes som dokumentasjon for senere benyttelse. Referatene bør i så fall ha en fast struktur slik at det er enkelt å forstå hvert referat. De bør også lagres på en åpen og oversiktlig måte slik at man ikke trenger å åpne en og en fil, men at man kan få en oversikt over alt innholdet samtidig. Det er viktig å huske på at dokumentasjon må holdes oppdatert for å oppfylle funksjonen om å informere om hva som er gjeldende. Hvis man begynner å tvile på dokumentasjonen, kan den miste sin nytteverdi og dermed kan tiden som legges inn i å holde den oppdatert være bedre brukt på andre ting.

En saksbehandler påpeker at kommunikasjonen utenom saksbehandlertreffene må være god. Videre sier vedkommende at man må vite hvor man skal henvende seg, og man må ha en ramme for hvordan spørsmålene kommer frem. Ved å ha standardiserte rammer for henvendelser, kan det være mulig å koble spørsmålene fra saksbehandlere sammen og dermed benytte dataen som beslutningstøtte i avgjørelser som skal tas. Det er ikke undersøkt om dette kun er et generelt utsagn eller om det er et reelt savn.

En fra utviklingsteamet foreslo å implementere et finkornet tilbakemeldingsverktøy der håpet er at man bruker mindre tid på kommunikasjonsbarrierer og mer tid på selve problemet. På den andre siden har en saksbehandler påpekt at kommunikasjonen utenom saksbehandlertreffene må være god, at saksbehandlerne må vite hvor de skal henvende seg og at de må ha en ramme for hvordan spørsmålene kommer frem. Et slikt verktøy vil gi en ramme for hvordan spørsmålene kommer frem, gitt at det er én standard gjennom hele systemet. Brukerne vil også da vite hvor de skal henvende seg, for det vil være mulig å gi tilbakemelding «over alt». Likevel bør det stilles spørsmål ved om dette finkorna tilbakemeldingsverktøyet faktisk gir mindre kommunikasjonsbarrierer eller ikke. Én ting er å kommunisere direkte bugge (feil i dataprogram) i systemet. En annen ting er å kommunisere at en funksjonalitet ikke fungerer slik den *bør*. Det kan hende funksjonaliteten fungerer slik den er *ment*, men at det som er ment viser seg å være feil. Dette må fortsatt kommuniseres på en tydelig måte til teamet slik at funksjonaliteten kan rettes opp på riktig vis. Mengden arbeid som kan ligge bak verktøyet kan være stort i forhold til om det faktisk gir mindre problemer vedrørende kommunikasjonsbarrierer, og man bør følgelig være forsiktig med å implementere dette.

6.5.2 Om supersaksbehandlerrollen

En saksbehandler uttrykker usikkerhet rundt hva som begrenser supersaksbehandlerne sitt ansvarsområde. Ved å ha en tydelig avklaring overfor saksbehandlerne om rammene til supersaksbehandlerne sitt domene, kan det i samme prosess gi saksbehandlerne en bedre forståelse for eget domene; kanskje også domenet til de forskjellige rollene i teamet, noe som kan gjøre det enklere for saksbehandlerne å kontakte teamet hvis det skulle være noe.

Utviklingsteammedlemmer trekker frem at når man blir bedre kjent med noen saksbehandlere, så gir det en annen type dialog enn den man har med de man ikke kjenner så godt. Dette kan utnyttes ved å inkludere både brukere man kjenner godt og brukere man ikke kjenner så godt. Dermed kan man få forskjellig type innspill. Hvis man «kun» snakker med de man kjenner godt, kan det medføre at man skjønner hva disse enkeltpersonene ønsker seg, selv om det ikke trenger å være representativt for resten av brukergruppa.

Rasmussen et al. (2011, p. 176) fant at brukere ofte blir valgt til å medvirke basert på om de kan bidra i fremdriften av IT-utviklingen heller enn om de er representative for brukergruppa av systemet. Ifølge et utviklingsteammedlem skal supersaksbehandlerne i den nye løsningen komme fra flere enheter i NAV fordi man gjør ting litt forskjellig fra enhet til enhet. Det er uvisst om ting gjøres forskjellig fordi man har forskjellig saksområde i de forskjellige enhetene eller om det er fordi forskjellige enheter vil behandle saker ulikt. Hvis det er sistnevnte bør dette tas tak i slik at saker blir behandlet likt uavhengig av hvilken enhet som behandler saken. Det er funnet at representasjonen av de enhetene som er mest ulike ikke er god nok.

Dette kan føre til en skjevhet i produktet. Utviklingsteamet ønsker allerede å få en bedre representasjon fra de forskjellige enhetene, som kan tyde på at de er bevisste på problemet. Rasmussen et al. (2011, p. 176) sier videre at det er viktig at det er de viktigste delene av brukergruppa som er representert. Kanskje anses de allerede representerte enhetene som de viktigste i utviklingen hos Team Bømlo; hvis ikke, er dette viktig å huske på i utvelgelsen av flere supersaksbehandlere.

En saksbehandler føler det er frustrerende at «alt» vedrørende utvikling av det nye systemet skal gjennom supersaksbehandlerne. Det gjennomføres blant annet saksbehandlertreff der saksbehandlerne ikke trenger å gå gjennom supersaksbehandlerne. Dette kan tyde på at utsagnet ikke er helt korrekt. Likevel er det interessant å finne ut om supersaksbehandlerleddet er mest gunstig eller ugunstig når det kommer til å kartlegge behov fra brukerne. På den ene siden vet supersaksbehandlerne litt mer om hva utviklingsteamet jobber med i det gitte øyeblikket. Dette kan gjøre det enklere for teamet da de slipper «unødvendige» henvendelser midt i utviklingen av noe annet. På den andre siden må det vurderes om det er supersaksbehandlerne sitt ansvar å vurdere hva som er viktig og ikke. Hvis det for eksempel er en stor gruppe med supersaksbehandlere, og alle disse får én eller to henvendelser om samme ting og dermed vurderer det som «ikke viktig», kan det føre til at viktige behov forblir urørt. Det er dermed viktig å gjøre et bevisst valg om hvem som skal vurdere viktigheten av problemstillinger som tas opp fra brukerne; er det supersaksbehandlerne, utviklingsteammedlemmer, eller er det brukerne selv. Denne avgjørelsen bør tas fra ledelsen over teamet selv.

En saksbehandler har fått til svar fra supersaksbehandlere at forslagene til vedkommende ikke er «vits å ta nå» eller at det er prøvd før, noe som er opplevd som frustrerende for vedkommende. Frustrasjon hos brukerne kan være uønsket, da det kan føre til gnisninger. Dette kan forebygges ved å ha bedre opplæring av supersaksbehandlerne om hvordan de skal kommunisere at «forslaget ikke er viktig akkurat nå» slik at saksbehandleren ikke blir frustrert, men får forståelse. Men, igjen er det viktig å være bevisst på om det er supersaksbehandlerne, utviklingsteammedlemmer eller saksbehandlerne som skal vurdere viktigheten av brukerne sine behov. Saksbehandlerne burde, som tidligere diskutert, ha mer direkte innvirkning på prioriteringer.

6.5.3 Saksbehandlertreff

Beck og Andres (2005, p. 82) sier at ved benyttelse av XP-prinsipper så er ikke rollene fastsatt og strenge. Man har et mål om at alle skal bidra så godt de kan for teamets suksess. Teammedlemmer fra Team Bømlo deltar på saksbehandlertreffene uavhengig av rolle. Teamet benytter treffene til å lære hvordan brukerne benytter systemet. Denne kunnskapen kan anses som relevant enten man er designer, utvikler eller jurist. Dermed kan alle teammedlemmer få en bedre forståelse av hvordan systemet bør bli, uavhengig av roller.

Beyer (2010, p. 18) hevder at brukere ikke klarer å gjengi arbeidsoppgavene sine detaljert hvis de blir spurt om å gjøre det. Dette kan tyde på at det kan være nyttig å observere brukere i arbeid i kombinasjon med samtaler der man får frem arbeidsoppgavene. Da kan observasjonen utfylle brukernes gjengivelse av arbeidsoppgavene sine slik at

utviklingsteammedlemmene får et mer helhetlig bilde. Dette underbygger dermed å gjennomføre en type observasjon av brukerne, slik som Team Bømlo gjør.

To fra utviklingsteamet anser at den største utfordringen ved treffene er at saksbehandlere kan føle seg «testet». Dersom man føler at det er en selv som blir testet så kan man vurdere ting på en litt annen måte enn hvis man ikke føler seg testet. Noen saksbehandlere har også sagt at de må minne seg selv på at det er systemet som skal observeres, og ikke saksbehandleren selv. Det blir nevnt at en løsning på dette kan være å gjennomføre flere treff med samme saksbehandler og at fra og med det andre treffet får begge parter mer ut av aktiviteten. Det kan være interessant å undersøke om denne tryggheten kan dyrkes frem på noe annet vis slik at begge parter kan få mer verdi allerede ut av første treff.

6.6 Teamets lærdom gjennom brukermedvirkning

Saeeda et al. (2020, p. 2) påpeker at hvis man involverer brukere som ikke har så mye kunnskap om hvilke kravspesifikasjoner som kan funke og ikke, så kan det medføre fiasko for de tekniske aspektene ved produktet. Dette kan underbygge utviklingsteamet sin fremgangsmåte med å lytte til brukerne, men innføre de funksjonalitetene de selv tror vil gi best resultat basert på egen kunnskap. Likevel har utviklingsteamet fått høre at de ikke har levert godt nok i forbindelse med automatisering i ny løsning, noe de anser som tungt å høre når de gjør så godt de kan. Et teammedlem anser dette som det vanskeligste med hele produktet. Robertson og Robertson (2013, p. xxi) legger vekt på hvor viktig det er å være sikker på at man løser det riktige problemet. Dermed er det interessant å se på følgende punkter.

- Brukerne bør ha mulighet til å forklare godt nok hvilke prosesser som kan automatiseres.
- Brukerne bør ha mulighet til å forklare godt nok hva som er nødvendig i den nye løsningen.
- Begge parter bør ha en god nok forventningsavklaring før og underveis i utviklingen slik at begge parter vet hva som forventes av hverandre og en selv.

Både det første og det andre punktet kan kobles opp mot at brukere har påpekt at de synes det er vanskelig at utviklingsteamet ikke skjønner hvorfor noe er viktig for dem. Ut ifra dette funnet kan det virke som at brukere føler at de har kommunisert så godt de kan, men at utviklingsteammedlemmer ikke forstår det som forsøkes kommunisert. Hvis teammedlemmer ikke har forstått det som kommuniseres, kan det vurderes til at brukerne ikke har forklart godt nok hva som er nødvendig i den nye løsningen, eller at de to partene ikke snakker «samme språk». Det er viktig at man snakker samme språk slik at begge parter forstår hva den andre parten forsøker å kommunisere. Begge disse første og andre punktene kan også sees i sammenheng med at brukere har påpekt at de føler at saksbehandlertreff er en «sjanse» de må gripe for å dele deres tanker rundt behov fra systemet. Dette kan tyde på at brukere føler de ikke kommer godt nok til for å ytre sine behov ellers i utviklingsprosessen. Det kan løses for eksempel ved at det innføres flere treff med hver bruker, at brukere bør få delt mer av sine meninger gjennom andre kanaler og at begge parter blir bedre kjent med den andre parten sine ord og uttrykk slik at de forstår hverandre.

Det tredje punktet, om begge parter har hatt god nok forventningsavklaring ovenfor hverandre slik at begge parter vet hva som forventes av hverandre og en selv, kan kobles opp til ønsket fra en saksbehandler om tydeligere ansvarsavklaring; dermed kan begge parter bli tryggere på hva man kan forvente av hverandre og hva som forventes av en selv. Tilbakemeldingen om at utviklingsteamet ikke har levert godt nok kan skyldes at ansvars- og forventningsavklaringen mellom de to partene ikke har vært tilstrekkelig.

«Folk lager artefakter som skal hjelpe dem med å gjennomføre arbeidet sitt. [...] Hvert artefakt gir innblikk i hvordan brukeren går frem for å gjøre arbeidet og hva de trenger for å støtte det.» (Beyer, 2010, p. 32). Et teammedlem har påpekt at man lærer mye av de tingene som brukerne benytter seg av utenfor systemet og utenfor skjermen; dette kan sammenlignes med artefakter. Artefaktene kan føre til at arbeidet blir gjennomført forskjellig fra hvordan systemet ville gjort det, og dermed også forskjellig fra hvordan det er ønsket (Beyer, 2010, p. 32). Det kan også føre til at saker blir behandlet forskjellig avhengig av vanene (eller uvanene) til hver enkelt saksbehandler. Dette er også påpekt fra teammedlemmet som en mulig situasjon som er ønsket å unngå. Det kan være nyttig å fange opp disse artefaktene (Beyer, 2010, p. 32) slik at man kan implementere dem i løsningen eller utnytte at de benyttes som eksterne hjelpemidler. Hvis man vet at de benyttes, kan teamet sørge for at resultatet av at noen benytter dem bli det samme som for de som ikke benytter dem. Artefaktene kan bli oppdaget gjennom samtaler og observasjon av brukerne i arbeid. Dette kan løses ved å innføre opplæring av de forskjellige artefaktene, eller ved å få funksjonaliteten til artefaktene implementert inn i systemet.

Robertson og Robertson (2013, p. xxi) legger vekt på hvor viktig det er å være sikker på at man løser det riktige problemet og at man må gjøre en bedre jobb på dette området ved utarbeidelse av kravspesifikasjoner. Et utviklingsteammedlem ga et eksempel der det hadde blitt brukt mye tid på en spesifikk ordlyd, men som likevel ble misvisende i forhold til hensikten. I dette tilfellet ville det blitt feil i utbetalingen til søker hvis det ikke hadde blitt oppdaget og rettet opp i. Dette kan vise at å bruke mye tid på noe ikke alltid gir et godt resultat. Videre kan dette være et tegn på at dokumentasjonen på hvordan ordlyden, altså behovet eller kravet fra bruker, skulle være ikke var tilstrekkelig. Beyer (2010, p. 41) sier at brukerhistorier bør inneholde høy-nivå beskrivelser av funksjonalitet eller oppførsel. Utviklingsteamet i NAV lager brukerhistorier med enkle beskrivelser av ønsket funksjonalitet eller oppførsel, slik som litteraturen anbefaler, og benytter seg ikke spesifikt av krav. «Oppgavetekst» og skisser endres flere ganger for at det skal være oppdatert. I denne prosessen kan noen av behovene, eller kravene, gå tapt hvis kravene ikke er dokumentert på ett sted.

Brukere ønsker å være hjelpsomme, noe som kan medføre at de responderer på småting i systemet istedenfor å få fram at de egentlig trenger et annet system, hvis det skulle være tilfellet (Beyer, 2010, p. 18). Dermed er det viktig å involvere brukere tidlig og avdekke *alle* brukernes behov; det kan gi misvisende resultater hvis man kun spør brukerne om «denne løsningen var fin» da det kan hende at løsningen er «fin», men feil i forhold til hva brukerne trenger. Utviklingsteammedlemmer har påpekt at brukerne ønsker å hjelpe. Hvis brukere tror at noe ikke er relevant enda eller at det allerede jobbes med, så kan de la være å dele feilen eller funksjonaliteten som trengs. Dette kan knyttes sammen med funnet om at saksbehandlere kan dele forskjellig type informasjon avhengig av hvor godt de kjenner

utviklingsteamet. Dette kan være lurt av utviklingsteamet å utnytte ved å ha god dialog både med saksbehandlere de kjenner godt og som har god kjennskap til utviklingsprosessen, og også med saksbehandlere de ikke kjenner så godt; både gjennom saksbehandlertreffene og gjennom annen kommunikasjon.

Det utvikles i et syntetisk miljø der dataene er litt annerledes enn de er i reelle sykepengesaker. Kontinuerlig testing er et viktig element i smidig utvikling (Meyer, 2018, p. 93). Dermed kan det være lurt å gjennomføre tester med reelle data, noe teamet allerede gjennomfører gjennom saksbehandlertreffene, monitorering, datapakker og oppfølging av det saksbehandlerne rapporterer inn.

Utviklingsteammedlemmer påpeker at det er viktig at utviklingsteamet får føle på kroppen hvordan systemet virker for de som saksbehandler i det, noe de opplever at skjer gjennom treffene. Saksbehandlere poengterer at treffene hjelper dem med å skjønne det nye systemet bedre. Begge disse funnene viser noen aspekter av hvordan begge parter ser seg tjent med å gjennomføre saksbehandlertreff. Gjennom observasjoner kan utviklingsteammedlemmer selv vurdere inntrykkene om hvordan systemet tas i bruk istedenfor å gå gjennom saksbehandleren sin vurdering gjennom samtaler. Vurderingen fra saksbehandleren kan være helt korrekt. Likevel kan det gi et sterkere inntrykk gjennom å observere det med egne øyne.

6.7 Terskel for å legge ut på Slack

Utviklingsteammedlemmer mener at det er lav terskel for å legge ut ting på Slack, både for brukere og for utviklingsteammedlemmer. Dette er for å få frem alle feil og mangler i systemet. Noen saksbehandlere har likevel presisert at terskelen oppleves som høy for å legge ut meldinger på Slack. Noen saksbehandlere har påpekt at de som regel må gjennom supersaksbehandlerne der de eventuelt får beskjed om å legge ut problemstillingen på Slack. Dette oppleves av disse saksbehandlerne å heve terskelen for å legge ut ting der. En saksbehandler er redd for at det vedkommende lurer på er dumt eller at man burde skjønne svaret selv. Dette kan skyldes at saksbehandleren ikke er like trygg på digitale plattformer fordi man ikke ser mottakeren sin reaksjon på det som skrives. Det kan også skyldes at det ikke har kommet godt nok frem at det ønskes en kultur der ingen spørsmål er for dumme, slik som er spesifisert av flere teammedlemmer gjennom intervjuer til dette studiet. I denne sammenheng nevner saksbehandlere også at taggesystemet i Slack er komplisert. Hvis de får en bedre innføring i taggesystemet, eventuelt jevnlig oppfriskninger, kan kanskje terskelen oppleves lavere for å legge ut spørsmål. Hvis terskelen for å legge ut ting på Slack er høyere for saksbehandlerne enn antatt av utviklingsteamet, kan dette medføre at behov fra brukerne ikke kommer frem til teamet så tidlig som teamet gjerne ønsker eller tror. Dette kan videre medføre at utviklingsteamet tror brukerne er mer fornøyde enn det de faktisk er; fordi de ikke hører fra dem. Dermed er et forslag til tiltak å ha bedre opplæring i bruk av taggesystemet i kommunikasjonskanalen Slack.

En saksbehandler sier at vedkommende opplever at man tilnærmet må ha bygget seg opp en «status» for å bli bedt av supersaksbehandler om å legge ut det vedkommende lurer på i Slack. Det er presisert fra utviklingsteammedlemmer at det er, og skal være, lav terskel for å legge ut ting på Slack. Dermed er det motstridende holdninger blant de to partene når det kommer til kommunikasjon via Slack. Det er observert at utviklingsteammedlemmer har en holdning om at hvis en sak blir meldt inn via Slack som ikke tas videre av teamet med en

gang, men saksbehandlerne mener at det haster, så vil det bli meldt inn på nytt ganske snart. Saksbehandlere har, som sagt, påpekt at de synes terskelen for å legge ut meldinger på Slack er høy til tider, noe som kan heves enda mer av utviklingsteammedlemmers holdning. Hvis supersaksbehandlere siler ut hva som kan spørres om i Slack, kan dette også medføre at ting som har blitt tatt opp tidligere ikke sendes inn på nytt fordi supersaksbehandlerne ser at det ikke ble tatt videre; likevel kan det hende at den gjeldende saken kan anses som *viktig nok* av teamet, hadde bare teamet fått en påminnelse. Dette kan føre til at utviklingsteamet tror at feil i systemet kommer frem tidligere enn de faktisk gjør. Dermed bør det vurderes om det er supersaksbehandlerne som skal vurdere hva som kan spørres om fra brukerne i Slack, eller om alle skal kunne spørre om hva enn de vil.

Det er observert noen tråder i Slack der sakene ikke virker å bli oppklart. I noen av disse tilfellene blir det sagt av utviklingsteammedlemmer at de skal komme tilbake til saksbehandler med mer informasjon, uten at dette er gjort over Slack. Det kan hende at denne informasjonen tar lenger tid å innhente enn observasjonsperioden til denne masteroppgaven har vært. Det kan også hende saksbehandleren har blitt kontaktet på en annen plattform som denne masteroppgaven ikke har innsyn i. Hvis det skulle være tilfellet at saker forblir uoppklart, kan det medføre at saksbehandlere er redd for at sin egen melding skal forbli ubesvart og dermed vegrer seg for å legge ut melding om noe de lurte på. Hvis dette de lurte på er vesentlig, så vil mest sannsynlig noen andre legge ut en melding på Slack om dette. Dermed kan det vurderes som at det er viktig at terskelen er lav nok til at feil i systemet kommer fram til teammedlemmer; hvis det krever mye arbeid for å få terskelen lav nok til at *alle* er komfortable med å legge ut ting på Slack, bør det vurderes om det er verdt det i forhold til om man får avdekket viktige feil i systemet slik det er nå. Det viktigste er at feil i, og behov fra, systemet kommer frem til utviklingsteamet.

6.8 Dokumentasjonskanaler

Når det lanseres ny funksjonalitet, legges dette ut i Slack med en beskrivelse av hvordan funksjonaliteten skal brukes. Det er ikke undersøkt om disse beskrivelsene lagres på en oversiktlig måte. Dermed er det usikkert om det er enkelt for saksbehandlerne, og utviklingsteammedlemmer, å finne tak i denne dokumentasjonen i senere tid; både når den fortsatt er gjeldende og når den eventuelt er utdatert.

Det dokumenteres gjennom koden til systemet og denne koden har versjonshåndtering. Det dokumenteres også hvilke paragrafer som benyttes til de forskjellige funksjonene og juristene kan, sammen med utviklere, gå inn i koden og verifisere at funksjonaliteten fungerer i henhold til lovverket. At juristene ikke kan verifisere funksjonalitet alene kan medføre at andre arbeidsoppgaver stopper opp fordi vedkommende må vente til en utvikler er tilgjengelig til å hjelpe til.

En saksbehandler ønsker mer dokumentasjon og oversikt over hvilke prosesser som foregår til enhver tid fra utviklingssiden og fra produkteiersiden. Saksbehandleren savner en god plattform der en slik oversikt kan kommuniseres og dokumenteres. Trello tas frem som et eksempel som finnes i dag, men at språket som benyttes der virker mer tiltenkt utviklingsteamet. Dermed kan det være en mulighet med en dokumentasjonskanal der saksbehandlerne også skjønner språket som blir brukt. En utfordring med å dokumentere hva man gjør til enhver tid, er å holde den oppdatert. Dette påpekes også av saksbehandleren

gjennom at noe som står som uferdig i dokumentasjonen faktisk kan være ferdig. En mulig løsning på dette kan være å benytte et språk i Trello som både gir mening for utviklingsteamet og for brukerne. Med dette kan det enten benyttes et annet språk enn det som benyttes i dag, eller så kan brukerne få opplæring i språket som benyttes av teammedlemmene.

7 Konklusjon og videre arbeid

7.1 Konklusjon

Konklusjonen besvarer problemstillingen og forskningsspørsmålene som er introdusert i kapittel 1. Delkapitlene er delt inn etter forskningsspørsmålene.

7.1.1 Hvordan anbefaler litteraturen at brukerne skal involveres i IT-utvikling?

Robertson og Robertson (2013, p. xxi) legger vekt på hvor viktig det er å være sikker på at man løser det riktige problemet og at man må gjøre en bedre jobb på dette området ved utarbeidelse av kravspesifikasjoner. Når man benytter XP-prinsipper, som er en smidig arbeidsmetode, så er ikke rollene fastsatt og strenge og man har et mål om at alle skal bidra så godt de kan for teamets suksess (Beck & Andres, 2005, p. 82).

Ved å involvere brukerne i utviklingen kan utviklingsteamet unngå å gjøre feil oppgaver (Liu & Wang, 2014, p. 1151). Litteraturen anbefaler at brukerne skal involveres kontinuerlig når man benytter smidige metoder (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32) og anses som en viktig del av det å jobbe smidig (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32; Yaman et al., 2016, p. 345). Det er «usikkerhet rundt fremtidige behov» i begynnelsen av et prosjekt med tanke på mål, leveranser og «fremtidige krav fra kunder og interessenter» (Johansen et al., 2019, pp. 56–57) som kan tyde på at å involvere brukerne tidlig i utviklingsprosessen kan være ettertraktet for å finne ut av dette. *Måten* brukerne blir involvert på er likevel viktig (Andersen, 2018, p. 410). Rasmussen et al. (2011, p. 176) fant at brukere ofte blir valgt til å medvirke basert på om de kan bidra i fremdriften av IT-utviklingen heller enn om de er representative for brukergruppa av systemet.

7.1.2 Hvordan involveres brukerne i utviklingen av systemet for behandling av sykepengesaker?

Utviklingsteamet utvikler gjennom kontinuerlig forbedring av produktet; det er en prosess som pågår uten slutt dato. Utviklingsteammedlemmer deltar på brukermedvirkningsaktiviteter uavhengig av rolle. Utviklingsteamet jobber stort sett digitalt, og de formelle brukermedvirkningsaktivitetene foregår følgelig også digitalt. Teamet gjennomfører ukentlige brukermedvirkningsaktiviteter, saksbehandlertreff, og ukentlige møter med alle saksbehandlere, og kun med supersaksbehandlere. De benytter også kontinuerlig brukermedvirkning; dette skjer gjennom kommunikasjon via en digital plattform, Slack, og å ha en åpen dør til brukerne. Teammedlemmer deler skisser med teamet og med brukerne og får tilbakemeldinger på disse. De sender også ut spørreundersøkelser til brukerne for å finne ut av hva teamet bør bli bedre på, eller på spesifikke ting i løsningen. Brukerne kan komme med innspill til forbedringer som teamet vurderer opp mot egen kunnskap og som de prioriterer deretter.

Teammedlemmer har fremhevet at det er viktig med tillit på tvers av teamet og mellom teamet og saksbehandlerne. Brukere har påpekt at de føler at de blir tatt seriøst, og at de er positivt overrasket over teamets evne til å sette seg inn i deres situasjon. Dette kan bidra til å øke tilliten mellom partene. Videre påpeker både saksbehandlere og teammedlemmer at de føler at saksbehandlertreff bidrar til å gi bedre forståelse for hvordan systemet er og hvorfor

det er sånn. Saksbehandlere har sagt at dette bidrar til å stole mer på systemet og «valgene» det tar i saksbehandlingen.

7.1.3 Hvilke brukermedvirkningsaktiviteter bør endres?

Organisering

Teammedlemmer uttrykker at både teammedlemmer og saksbehandlere har et stort arbeidspress. Teammedlemmene føler seg presset fremover der de stadig må levere ny funksjonalitet. Saksbehandlere måles på antall saker, noe som kan føre til at de føler et enda større arbeidspress. Det er viktig at de riktige KPI-ene velges slik at organisasjonen får frem den ønskelige arbeidsatferden; for eksempel gjennom å måle antall saker som blir behandlet *feil* gjennom den nye løsningen i forhold til lovverket, eller antall situasjoner der saksbehandlere ikke skjønner hva som menes i den nye løsningen.

Organiseringen av brukermedvirkningsaktiviteter virker å være relativt avhengig av enkeltpersoner slik det er lagt opp i dag. Brukermedvirkning anses som en viktig del av det å jobbe smidig, både fra intervjuobjekter og fra litteraturen (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32; Yaman et al., 2016, p. 345). Dermed bør brukermedvirkningsaktiviteter organiseres slik at de er en del av arbeidskulturen og ikke avhenger av enkeltpersoner for å bli gjennomført.

Det er funnet at saksbehandlere omtaler utviklingen som et *prosjekt* mens teammedlemmer presiserer at arbeidsmetoden er *produktbasert utvikling* der det ikke er noe slutt-tidspunkt for utviklingen. Det bør vurderes om disse uklarhetene i forventninger om arbeidsmetode påvirker fremgangen i utviklingen og løsningene for produktet. Hvis dette er tilfellet, bør gjeldende arbeidsmetode klareres.

Teammedlemmer påpeker at tiden fra man har lært noe nytt fra brukerne til teamet får implementert nødvendige endringer i systemet er for lang. Dermed kan teamet miste en viktig fordel av å jobbe smidig sammen med kontinuerlig brukerinvolvering. Kontinuerlig brukerinvolvering er en viktig del av å jobbe smidig (Inayat et al., 2012, p. 1; Schmitz et al., 2019, p. 32) som man ikke får utnyttet til det fulle hvis man ikke får implementert ny lærdom inn i systemet.

Det er et ønske fra teammedlemmer og saksbehandlere å bedre kommunikasjonen vedrørende fremdriften til utviklingen. Et teammedlem, i tråd med Iivari og Molin-Juustila (2009, p. 55), påpeker at man bør anse brukerne og utviklerne som fagpersoner der man benytter hver sin fagkunnskap i utviklingen. Denne fremgangsmåten kan være et tiltak for å bedre den kontinuerlige kommunikasjonen med brukerne.

Supersaksbehandlerne i utviklingen hos Team Bømlø er ansett av teammedlemmer som å ikke være representative for hele brukergruppa. Supersaksbehandlergruppa skal utvides, så representasjon av hele brukergruppa bør være i fokus i utvelgelse av nye medlemmer. Videre er det i dag lagt opp til at brukerne aktivt sier ifra om ting. Dermed bør det vurderes om denne fremgangsmåten fanger opp den gjennomsnittlige brukeren sine synspunkter, eller om det kun er de engasjerte brukere sine meninger som kommer frem. Videre påpeker en saksbehandler at vedkommende opplever det som frustrerende at «alt» vedrørende utvikling av det nye systemet skal gjennom supersaksbehandlerne. Det er viktig at utviklingsteamet

gjør et bevisst valg om hvem som skal vurdere viktigheten av problemstillinger som tas opp fra brukerne i benyttelsen av supersaksbehandlere; er det supersaksbehandlerne, utviklingsteammedlemmer eller brukerne selv. Denne avgjørelsen bør tas fra ledelsen over teamet selv.

Kommunikasjon

Det viktigste ved softwareutvikling er verdien det gir til eierne eller brukerne (Robertson & Robertson, 2013, p. xxi). Både teammedlemmer og saksbehandlere har påpekt at de synes det er vanskelig å kommunisere ovenfor den andre parten hvorfor noe anses som viktig. En løsning på dette kan være at brukerne blir bedre informert om de store linjene i utviklingen og at de får mer direkte innflytelse på prioriteringer.

Et teammedlem foreslår implementering av et finkornet tilbakemeldingsverktøy som skal gjøre at man kan bruke mindre tid på kommunikasjonsbarrierer og mer tid på selve problemet. Om dette faktisk blir effekten av et slikt verktøy er diskuterbart; mengden arbeid som kan ligge bak verktøyet kan være stort i forhold til vinning, og man bør være forsiktig med å implementere dette.

Digital skriftlig kommunikasjon, Slack

Det er et ønske fra teammedlemmer at det skal være lav terskel for saksbehandlerne å stille spørsmål i Slack; terskelen er ønsket lav for å få frem alle oppdagede feil og mangler i systemet. Saksbehandlere har påpekt at de synes terskelen for å legge ut på Slack er høy. En grunn som nevnes for dette er at taggesystemet er for avansert. Hvis det krever mye arbeid for å få terskelen lav nok til at *alle* er komfortable med å legge ut ting på Slack, bør det vurderes om det er verdt det i forhold til om man får avdekket viktige feil i systemet slik det er nå. Et forslag til tiltak er å ha bedre opplæring i bruk av taggesystemet og vurdere hvor lav terskelen for å legge ut ting på Slack trenger å være.

En saksbehandler sier at supersaksbehandlere vurderer hva som kan legges ut på Slack fra brukerne sin side. Dersom supersaksbehandlerne skal vurdere viktigheten av problemer og spørsmål kan det føre til at utviklingsteamet tror at feil i systemet kommer frem tidligere enn de faktisk gjør. Dermed bør det vurderes om supersaksbehandlerne skal vurdere hva som kan spørres om fra brukerne i Slack, eller om alle skal kunne spørre om hva enn de vil.

Saksbehandlertreff

Saksbehandlertreff anses av intervjuobjektene å være en fin arena der teammedlemmer selv kan se hvordan systemet oppfører seg. Dette kan være fint i kombinasjon med samtaler med brukerne om deres behov. Likevel har flere, både teammedlemmer og saksbehandlere, påpekt at saksbehandlere kan føle seg testet under treffene og dermed tar valg på en annen måte enn de ville gjort ellers; denne effekten kan minskes ved at saksbehandlere virkelig får en forståelse av at det er systemet som undersøkes, ikke saksbehandleren selv. Dette kan gjennomføres ved at de deltar på flere treff.

Dokumentasjon og feiltesting

Dokumentasjon for fremdrift og valg av løsninger finnes i koden til systemet, på Slack eller gjennom Trello der utviklingsteamet legger inn oppgaver som må gjøres. Det bør innføres en mer oversiktlig dokumentasjon over hvordan utviklingen ligger an.

Teammedlemmer har funnet ut at noe funksjonalitet de har brukt mye tid på, endte opp med å gi feil resultat. Dette kan tyde på at det er et behov for bedre dokumentasjon av brukernes behov.

En saksbehandler påpeker at gamle feil kommer tilbake igjen i systemet. Et forslag til forbedring er at det ved all feilretting bør legges inn automatiske tester som feiler hvis den kjente feilen skulle gjenoppstå, og kommunikasjonen ut til brukerne om når en feil er rettet bør bli bedre.

7.2 Videre forskning

Hvis studiet skulle blitt gjentatt anbefales det å gjennomføre flere intervjuer om mer generell brukermedvirkning i teamet. Det ble stilt en del generelle oppfølgingsspørsmål ved de spesifikke intervjuene, likevel anbefales det å gjennomføre systematiske generelle intervjuer om brukermedvirkning med både saksbehandlere og teammedlemmer i flere roller.

Videre anbefales det å undersøke om brukerne får uttrykt sine behov i tilstrekkelig grad til utviklingsteamet. Brukere har påpekt at de anser det vanskelig å kommunisere hvorfor noe er viktig for dem; dermed kan det videre undersøkes hvordan kommunikasjonen blir uttrykt fra brukerne og hvordan teammedlemmer mottar det.

Det anbefales å undersøke hvordan verdien fra uformelle samtaler kan beholdes ved et heldigitalt team; for eksempel prat om tregheter ved kaffemaskinen i en pause. Kanskje dette kan videreføres gjennom digitale kaffepauser, for eksempel på ti minutter, for alle på fast tidspunkt hver dag?

Et utviklingsteammedlem sier at man gjør ting litt forskjellig fra enhet til enhet i NAV. Det er ikke forsket på om ting gjøres forskjellig fordi man har forskjellig saksområde i de forskjellige enhetene eller om det er fordi forskjellige enheter vil behandle saker ulikt. Dette bør undersøkes og dersom det er sistnevnte bør dette tas tak i slik at saker blir behandlet likt uavhengig av hvilken enhet som behandler saken.

Det kan være interessant å undersøke om tryggheten som oppstår etter å ha deltatt på flere saksbehandlertreff kan komme frem på noe annet vis; dermed kan begge parter få mer verdi allerede ut av første treff.

Det kan være spennende å undersøke hvorfor brukere velger å ikke være aktive i brukermedvirkningsaktiviteter; spesifikt ved utviklingsprosesser der medvirkningen avhenger av brukere som er aktive i denne prosessen.

Det anbefales å gjennomføre observasjon av saksbehandlertreff slik at forskeren selv kan danne seg inntrykk av hvordan dette foregår. Det er interessant å undersøke implementeringen av konklusjonene fra denne oppgaven og hvordan dette påvirker brukertilfredsheten og systemet.

Kilder

- A/B testing*. (2022, February 14). Optimizely. <https://www.optimizely.com/no/optimization-glossary/ab-testing/>
- Abelein, U., & Paech, B. (2014). State of Practice of User-Developer Communication in Large-Scale IT Projects. In C. Salinesi & I. van de Weerd (Eds.), *Requirements Engineering: Foundation for Software Quality* (pp. 95–111). Springer International Publishing. https://doi.org/10.1007/978-3-319-05843-6_8
- AcademicWork. (2021, July 12). *Semistrukturert intervju og andre teknikker*. Academic Work. <https://www.academicwork.no/insights/arbeidsgivere/intervjuteknikker>
- Adam, J. (2021, July 26). *What is Agile software development? - K&C*. <https://kruschecompany.com/agile-software-development/>
- Admin. (2020, June 23). *What is Agile Methodology? Agile Software Development*. <https://www.zucisystems.com/software-development/agile-methodology/>
- Agile Alliance. (2017, June 14). What is Extreme Programming (XP)? *Agile Alliance* |. <https://www.agilealliance.org/glossary/xp/>
- Andersen, E. S. (2018). *Prosjektledelse—Et organisasjonsperspektiv* (2. utgave). Fagbokforlaget.
- Beck, K., & Andres, C. (2005). *Extreme Programming Explained* (second edition).
- Beyer, H. (2010). *User-Centered Agile Methods*. Morgan & Claypool Publishers.
- Bosch-Sijtsema, P., & Bosch, J. (2015). User Involvement throughout the Innovation Process in High-Tech Industries. *Journal of Product Innovation Management*, 32(5), 793–807. <https://doi.org/10.1111/jpim.12233>
- Civil Service India. (2021, July 10). *User Involvement, User Involvement in System Development*. <https://www.civilserviceindia.com/subject/Management/notes/user-involvement.html>
- Dalen, M. (2022, May 23). *Validitet og reliabilitet i kvalitativ forskning*.

- Dingsøy, T., Moe, N., Fægri, T., & Amdahl Seim, E. (2018). Exploring software development at the very large-scale: A revelatory case study and research agenda for agile method adaptation. *Empirical Software Engineering*, 23, 1–31. <https://doi.org/10.1007/s10664-017-9524-2>
- Eriksen, K. F. (2022, March 29). *Smidig-problemer: Hva gjør du når noen alltid tar de morsomste oppgavene?* Digi.No. <https://www.digi.no/artikler/ny-ntnu-forskning-fire-steder-skoen-trykker-for-smidige-team/518340>
- Gothelf, J., & Seiden, J. (2016). *Lean UX: Designing Great Products with Agile Teams*. O'Reilly Media, Inc.
- Grunnbeløpet i folketrygden*. (2021, May 21). nav.no. <https://www.nav.no/no/nav-og-samfunn/kontakt-nav/utbetalinger/grunnbelopet-i-folketrygden>
- Hoda, R., Salleh, N., & Grundy, J. (2018). The Rise and Evolution of Agile Software Development. *IEEE Software*, 35(5), 58–63. <https://doi.org/10.1109/MS.2018.290111318>
- Iivari, N., & Molin-Juustila, T. (2009). Listening to the Voices of the Users' in Product Based Software Development. *International Journal of Technology and Human Interaction*, 5(3), 54–77. <https://doi.org/10.4018/jthi.2009070103>
- Inayat, I., Noor, M. A., & Inayat, Z. (2012). Successful product-based agile software development without onsite customer an industrial case study. *International Journal of Software Engineering and Its Applications*, 6(2), 1–14. Scopus.
- inkluderingsdepartementet, A. (2006, June 17). *Arbeids- og velferdsetaten (NAV) [Eksternvirksomhet]*. Regjeringen.no; regjeringen.no. https://www.regjeringen.no/no/dep/aid/om-arbeids-og-inkluderingsdepartementet/etatstyring/underliggende-etater/arbeids_og_velferdsetaten/id1511/

- Intro til universell utforming.* (2022, February 14). uutilsynet.no.
<https://www.uutilsynet.no/brukerutfordringer/intro-til-universell-utforming/238>
- Introduction to Agile Methodology: Comprehensive Explanation » Network Interview.* (2020, October 21). <https://networkinterview.com/introduction-to-agile-methodology/>
- Jacobsen, D. I. (2021). *Hvordan gjennomføre undersøkelser? Innføring i samfunnsvitenskapelig metode.*
- Johansen, A., Olsson, N. O. E., Jergeas, G., & Rolstadås, A. (2019). *Project Risk and Opportunity Management—An Owner’s Perspective.* Routledge.
- Kerzner, H. (2017). *Project Management Metrics, KPIs, and Dashboards: A Guide to Measuring and Monitoring Project Performance.* John Wiley & Sons.
- Kniberg, H., & Ivarsson, A. (2012). *Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds.* 14.
- Kva er NAV?* (2022, January 10). nav.no. <https://www.nav.no/no/nav-og-samfunn/om-nav/fakta-om-nav/kva-er-nav>
- Lamelas, A. (2018, October 11). Top 5 main Agile methodologies: Advantages and disadvantages. *Xpand IT.* <https://www.xpand-it.com/blog/top-5-agile-methodologies/>
- Ledige stillinger—Arbeidsplassen.* (2022, March 3). <https://arbeidsplassen.nav.no/stillinger/stilling/be8a0630-64a4-44aa-9ad0-7d2aa2b84acb>
- Lee, J. (2022, February 3). *Spotify’s Failed #SquadGoals* [Blog]. Jeremiahlee.Com. <https://www.jeremiahlee.com/posts/failed-squad-goals/>
- Liu, S. (2016). How the user liaison’s understanding of development processes moderates the effects of user-related and project management risks on IT project performance. *Information & Management*, 53(1), 122–134. <https://doi.org/10.1016/j.im.2015.09.004>

- Liu, S., & Wang, L. (2014). User Liaisons' Perspective on Behavior and Outcome Control in IT Projects: Role of IT Experience, Behavior Observability, and Outcome Measurability. *ResearchGate*, 1148–1173. <https://doi.org/10.1108/MD-08-2013-0430>
- Medelyan, A. (2019, October 11). *Coding Qualitative Data: How To Code Qualitative Research (2021)*. Thematic. <https://getthematic.com/insights/coding-qualitative-data/>
- Meyer, B. (2018). Making Sense of Agile Methods. *IEEE Software*, 35(2), 91–94. <https://doi.org/10.1109/MS.2018.1661325>
- Mohagheghi, P., & Jørgensen, M. (2017). What Contributes to the Success of IT Projects? Success Factors, Challenges and Lessons Learned from an Empirical Study of Software Projects in the Norwegian Public Sector. *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, 371–373. <https://doi.org/10.1109/ICSE-C.2017.146>
- Nettsteder | Tilsynet for universell utforming av ikt. (2022, April 26). <https://www.uutilsynet.no/wcag-standarden/nettsteder/711>
- Olsen, D. (2015). *The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback*. John Wiley & Sons.
- Olseng, E. T., & Sundbye, L. M. T. (2021, October 25). *Observasjon, eksperiment og dybdeintervju—Markedsføring og ledelse 1—NDLA*. <https://ndla.no/nb/subject:1:47678c7b-bc09-4fc8-b2d9-a2e3d709e105/topic:1:dbdbb73b-2acb-49ac-bbce-b13d2aa409fd/resource:35f48ad2-0e29-4201-83fe-1be598982c5b>
- Organisering av NAV. (2022, January 5). nav.no. <https://www.nav.no/no/nav-og-samfunn/om-nav/fakta-om-nav/organisering-av-nav>
- Power, DanielJ. (2008). Understanding Data-Driven Decision Support Systems. *Information Systems Management*, 25(2), 149–154. <https://doi.org/10.1080/10580530801941124>

- Principles behind the Agile Manifesto.* (2021, November 29).
<https://agilemanifesto.org/principles.html>
- Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., Dar, S. H., & Khalid, S. (2021). Requirement Engineering Challenges in Agile Software Development. *Mathematical Problems in Engineering, 2021*, 1–18.
<https://doi.org/10.1155/2021/6696695>
- Rasmussen, R., Christensen, A. S., Fjeldsted, T., & Hertzum, M. (2011). Selecting users for participation in IT projects: Trading a representative sample for advocates and champions? *Interacting with Computers, 23*(2), 176–187.
<https://doi.org/10.1016/j.intcom.2011.02.006>
- Redström, J. (2005). *Towards user design? On the shift from object to user as the subject of design* (p. 14). Interactive Institute.
- Robertson, J., & Robertson, S. (2013). *Mastering the requirements process* (Third). Pearson Education.
- Rosenberg, D., Boehm, B., Stephens, M., Suscheck, C., Dhalipathi, S. R., & Wang, B. (2020). Test Early, Test Often. In D. Rosenberg, B. Boehm, M. Stephens, C. Suscheck, S. R. Dhalipathi, & B. Wang (Eds.), *Parallel Agile – faster delivery, fewer defects, lower cost* (pp. 107–130). Springer International Publishing. https://doi.org/10.1007/978-3-030-30701-1_6
- Saeeda, H., Dong, J., Wang, Y., & Abid, M. A. (2020). A proposed framework for improved software requirements elicitation process in SCRUM: Implementation by a real-life Norway-based IT project. *Journal of Software: Evolution and Process, 32*.
<https://doi.org/10.1002/smr.2247>
- Saksbehandler.* (2022, March 16). Utdanning.no.
<https://utdanning.no/yrker/beskrivelse/saksbehandler>
- Samset, K. (2015). *Prosjekt i tidligfasen—Valg av konsept* (2. utgave). Fagbokforlaget.

- Sander, K. (2019, November 30). Validitet. *eStudie.no*. <https://estudie.no/validitet/>
- Sander, K. (2020, November 22). Forskningsdesign. *eStudie.no*. <https://estudie.no/hva-er-forskningsdesign/>
- Schmitz, K., Mahapatra, R., & Nerur, S. (2019). User Engagement in the Era of Hybrid Agile Methodology. *IEEE Software*, 36(4), 32–40. <https://doi.org/10.1109/MS.2018.290100623>
- Swanson, R. A., & Holton, E. F. (2005). *Research in Organizations: Foundations and Methods in Inquiry*. Berrett-Koehler Publishers.
- Sykepengere til arbeidstakere. (2022, March 30). nav.no. <https://www.nav.no/no/person/arbeid/sykmeldt-arbeidsavklaringspenger-og-yrkesskade/sykepengere/sykepengere-til-arbeidstakere>
- Tatale, S., & Chandra Prakash, V. (2020). Enhancing acceptance test driven development model with combinatorial logic. *International Journal of Advanced Computer Science & Applications*, 11(10), 268–278. <https://doi.org/10.14569/IJACSA.2020.01111036>
- Universell utforming av apper | Tilsynet for universell utforming av ikt. (2022, April 26). <https://www.uutilsynet.no/regelverk/universell-utforming-av-apper/230>
- University of Mary Washington. (2016, May 31). *IT Project Management Standard*. Information Technologies. <https://technology.umw.edu/it-policies/project-management-standard/>
- WCAG sortert etter tema | Tilsynet for universell utforming av ikt. (2022, April 26). <https://www.uutilsynet.no/wcag-standard/wcag-sortert-etter-tema/729>
- What is Scrum? (2021, November 27). Scrum.Org. <https://www.scrum.org/resources/what-is-scrum>
- What is Usability Testing? (And What it Isn't) | Hotjar. (2022, February 14). <https://www.hotjar.com/usability-testing/>

- Yaman, S. G. (2018). User Involvement in Continuous Experimentation. *ACM SIGSOFT Software Engineering Notes*, 43(1), 1–4. <https://doi.org/10.1145/3178315.3178322>
- Yaman, S. G., Fagerholm, F., Munezero, M., Münch, J., Aaltola, M., Palmu, C., & Männistö, T. (2016). Transitioning Towards Continuous Experimentation in a Large Software Product and Service Development Organisation – A Case Study. In P. Abrahamsson, A. Jedlitschka, A. Nguyen Duc, M. Felderer, S. Amasaki, & T. Mikkonen (Eds.), *Product-Focused Software Process Improvement* (pp. 344–359). Springer International Publishing. https://doi.org/10.1007/978-3-319-49094-6_22
- Yaman, S. G., Munezero, M., Münch, J., Fagerholm, F., Syd, O., Aaltola, M., Palmu, C., & Männistö, T. (2017). Introducing continuous experimentation in large software-intensive product and service organisations. *Journal of Systems and Software*, 133, 195–211. <https://doi.org/10.1016/j.jss.2017.07.009>
- Yarbrough, Q. (2021, May 18). *Project Definition: The Meaning of a Project*. ProjectManager.Com. <https://www.projectmanager.com/blog/project-definition>
- Yin, R. K. (2018). *Case study research and applications—Design and methods*.
- Zhu, L., Johnsson, C., Varisco, M., & Schiraldi, M. M. (2018). Key performance indicators for manufacturing operations management – gap analysis between process industrial needs and ISO 22400 standard. *Procedia Manufacturing*, 25, 82–88. <https://doi.org/10.1016/j.promfg.2018.06.060>

Vedlegg

A. Observasjonsplan

Observasjoner og kommentarer ble skrevet ned fortløpende i kolonne B.

	A	B
1	Observasjonsskjema	
2	Prosjekt:	Saksbehandling av søknad til sykepenger
3	Team:	Team Bømlo
4	Forberedelser: Teamets situasjon	
5	Fysiske vs digitale møter:	
6	Deltakere i teamet: (oppgi hvor mange med hver stilling, typ utvikler osv)	
7	Antall brukere av systemet:	
8	Type produkt:	
9	Prosjektfase:	
10	Hvordan foregår kommunikasjonen mellom utviklere og brukere?	
11	Hvordan har covid-situasjonen med mer digitalt og hjemmekontor påvirket kommunikasjonen mellom brukerne og utviklingsteamet?	
12	Hvordan er arbeidsflyten/prosessene i teamet?	
13	Hvordan er arbeidsflyten/prosessene fra teamet til organisasjonen?	
14	Hvordan er teamets autonomitet?	
15	Arbeidsmetode til teamet? Typ Scrum, Kanban, Lean, Spotifymodellen? En blanding?	
16	Forberedelser: Gjennomføring av kommunikasjon med brukere	
17	Hvordan gjennomføres brukerinteraksjonen? Typ, direkte spørsmål, observasjon av at brukerne benytter systemet, osv?	
18	Hvordan lagres informasjonen fra brukermidvirkningen?	
19	Hvordan er representasjonen av brukerne i medvirkningsprosessene? Er hele brukergruppa representert?	
20	Observasjon av prosjektteamet	
21	Er kommunikasjonen fra utviklingsteamet til brukerne tydelig? Utdyp	
22	Hvordan er språkbruken fra utvikler?	
23	Bli sakene oppklart fra utvikler sin side?	
24	Går det mye kommunikasjon frem og tilbake for å løse/kommunisere et problem?	
25	Skjer lansering av nye løsninger på en ryddig måte?	
26	Bli brukerne hørt i sakene de tar opp?	
27	Hvordan er hurtigheten i svar fra utviklingsteamet til saksbehandlerne	
28	Overordnet: Virker det som at utviklerne skjønner brukerne sine behov?	
29	Observasjon av brukerne av systemet	
30	Virker det som at brukerne forstår det som blir kommunisert fra utviklingsteamet?	
31	Hvordan er språkbruken fra brukerne?	
32		

B. Intervjuguide generelt om brukermedvirkningsprosesser

Intervjuguide

- Individuelt, semistrukturert intervju
- Varighet ca. 60 min
- Dette intervjuet er én del av et større studium av brukermedvirkningsaktivitetene som gjennomføres av utviklingsteamet i NAV

Tema

Gjennomføring av brukermedvirkningsaktiviteter i et utviklingsteam som benytter smidig metode i NAV.

Problemstilling og forskningsspørsmål

Problemstillingen er foreløpig formulert til:

«Hvordan gjennomføres brukermedvirkningsaktiviteter i utviklingsteam i NAV, og hvordan kan aktivitetene forbedres?»

Og følgende forskningsspørsmål ønskes besvart.

1. Hvordan gjennomføres brukermedvirkningsaktivitetene mellom utviklingsteamet og deres brukere i dag?
2. Hvilke problemer oppstår ved brukermedvirkningsaktivitetene?
3. Hvordan kan brukermedvirkningsprosessen bedres for utviklingsteamet og brukerne?

Form

Intervjuet starter med en kort introduksjon, hvor intervjuer informerer intervjuobjekt om hva prosjektet omhandler og problemstillingen før intervjuobjekt får mulighet til å presentere seg selv. Intervjuer leder så intervjuobjekt gjennom intervjuet som vil bestå av noen nedskrevne spørsmål, samt. oppfølgingsspørsmål som stilles underveis. Intervjuet vil bli tatt opp på lydbånd via telefon eller video via Zoom/Teams.

1. Rammesetting

Uformell samtale: Informasjon om bakgrunn til prosjektet og problemstilling

Forklare hva intervjuet skal brukes til

Avklare eventuelle spørsmål intervjuobjekt har, f.eks. rundt anonymitet, taushetsplikt og andre spørsmål

Informere intervjuobjekt om at intervjuet blir tatt opp

Spør om det er noe som er uklart

2. Oppklaring

Overgangsspørsmål: Oppklaring

- Hva legger du i begrepet brukermedvirkning?
- Hva legger du i begrepet *smidig metode*?
- Hvordan er holdningene til brukermedvirkning i din bedrift og i teamet?
- Hvordan er motivasjonen for å gjennomføre brukermedvirkning i prosjektet deres?

3. Fokusering

Nøkkelspørsmål og oppfølgingsspørsmål

- Hvilke av brukermedvirkningsaktivitetene som gjennomføres i dag synes du bidrar i **størst** grad til å levere verdi tilbake til brukerne? Oppgi gjerne flere aktiviteter.
 - o Hvorfor?
- Hvilke av brukermedvirkningsaktivitetene som gjennomføres i dag synes du bidrar i **minst** grad til å levere verdi tilbake til brukerne? Oppgi gjerne flere aktiviteter.
 - o Hvorfor?
- Har det oppstått problemer i forbindelse med brukermedvirkningsaktivitetene? Hvis ja, hvilke?
 - o Hvordan ble disse problemene løst?
 - o Hvordan kunne problemene ha blitt løst på en bedre måte?
 - o Hva tror du kunne ha blitt gjort bedre for at disse problemene ikke hadde oppstått med det første?
- Har utviklingsteamet utforsket forskjellige brukermedvirkningsaktiviteter for å se hva som fungerer best?
 - o Hvis ja, hvilke aktiviteter og hvordan ble det utforsket?
- Hvordan benytter utviklerne dagen sin i forhold til arbeid med brukere vs andre aktiviteter?

4. Tilbakeblikk

Oppsummering

Avklare eventuelle misforståelser

Om ønskelig kan intervjuobjekt legge til noe mer, som intervjuer ikke har spurt om.

C. Intervjuguide spesifikt til observasjonsaktiviteten

Intervjuguide

- Individuelt, semistrukturert intervju
- Varighet ca. 30 min
- Dette intervjuet er én del av et større studium av brukermedvirkningsaktivitetene som gjennomføres av utviklingsteamet i NAV

Tema

Gjennomføring av brukermedvirkningsaktiviteter i et utviklingsteam som benytter smidig metode i NAV. Spesifikt for dette intervjuet er observasjon av saksbehandlere i arbeid.

Problemstilling og forskningsspørsmål

Problemstillingen er foreløpig formulert til:

«Hvordan gjennomføres brukermedvirkningsaktiviteter i utviklingsteam i NAV, og hvordan kan aktivitetene forbedres?»

Og følgende forskningsspørsmål skal besvares.

1. Hvordan gjennomføres brukermedvirkningsaktivitetene mellom utviklingsteamet og deres brukere i dag?
2. Hvilke problemer oppstår ved brukermedvirkningsaktivitetene?
3. Hvordan kan brukermedvirkningsprosessen bedres for utviklingsteamet og brukerne?

Form

Intervjuet starter med en kort introduksjon, hvor intervjuer informerer intervjuobjekt om hva prosjektet omhandler og problemstillingen før intervjuobjekt får mulighet til å presentere seg selv. Intervjuer leder så intervjuobjekt gjennom intervjuet som vil bestå av noen nedskrevne spørsmål, samt. oppfølgingsspørsmål som stilles underveis. Intervjuet vil bli tatt opp på lydbånd via telefon eller video via Zoom/Teams.

1. Rammesetting

Uformell samtale: Informasjon om bakgrunn til prosjektet og problemstilling

Forklare hva intervjuet skal brukes til

Avklare eventuelle spørsmål intervjuobjekt har, f.eks. rundt anonymitet, taushetsplikt og andre spørsmål

Informere intervjuobjekt om at intervjuet blir tatt opp

Spør om det er noe som er uklart

2. Oppklaring

Overgangsspørsmål: Oppklaring

- Er du saksbehandler eller utviklingsteammedlem?

3. Fokusering

Nøkkelspørsmål og oppfølgingsspørsmål

- Hvordan gjennomføres observasjonen?
- Blir det oppdaget feil i systemet gjennom observasjonene?
- Har møtene en klar agenda?
- Er kommunikasjonen tydelig fra utvikler/saksbehandler under observasjon/møtet?
- Føler du det har oppstått problemer under observasjonene mellom deg og utvikler/saksbehandler? Hvis ja, hvilke?
 - o Hvordan ble disse problemene løst?
 - o Hvordan kunne problemene ha blitt løst på en bedre måte?
 - o Hva tror du kunne ha blitt gjort bedre for at disse problemene ikke hadde oppstått med det første?
- Føler du at observasjon/møte med utvikler/saksbehandler er nyttig?
 - o Hvorfor/hvorfor ikke?
 - o Hvis det ikke er nyttig, hva kan gjøres for at det skal bli mer nyttig?
 - o Hvilke andre aktiviteter føler du vil gi mer nytte til brukermedvirkningsprosessen?

4. Tilbakeblikk

Oppsummering

Avklare eventuelle misforståelser

Om ønskelig kan intervjuobjekt legge til noe mer, som intervjuer ikke har spurt om.

D. Det smidige manifest

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Principles behind the Agile Manifesto

We follow these principles:

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Business people and developers must work together daily throughout the project.

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Continuous attention to technical excellence and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

E. De syv prinsippene for brukersentrert smidig utvikling i NAV



Prinsipp 1

Vi setter oss inn i brukerens situasjon

Vi tar utgangspunkt i det vi vet om brukerne og øker
kontinuerlig brukerinnsikten

Ved å sørge for at vi får raske tilbakemeldinger fra brukeren
forstår vi behovet og hvordan vi kan forbedre produktet.

Vi bruker tilgjengelige og lovlige metoder for datainnsamling





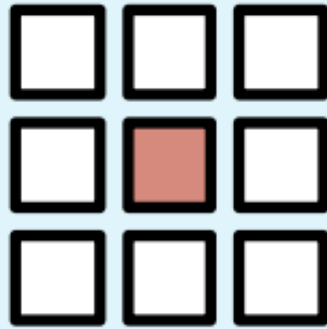
Prinsipp 2

Vi realiserer produktområdets mål

Produktområdet setter mål og retning for produktteamene.

Teamene definerer og prioriterer oppgaver som gradvis realiserer målbildet.





Prinsipp 3

Vi er selvstyrte... ...innenfor gitte rammer

Vi ønsker selvstyrte team som tar gode beslutninger raskt, for å oppnå høyere fart og bedre kvalitet. Dette gjøres mest effektivt i tverrfaglige team som forstår vår virksomhet og har nødvendig beslutningsmyndighet for sitt produkt.

Teamet forholder seg til retningslinjer og skal etterleve relevante krav.





Prinsipp 4

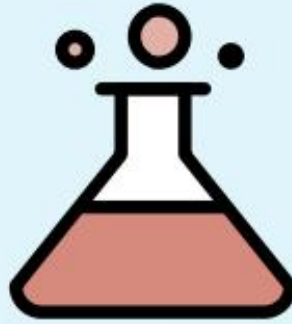
Vi leverer verdi ofte

Vi leverer små endringer ofte for å få rask tilbakemelding fra brukerne - slik verifiserer vi verdi og kvalitet fortløpende.

Vi eier vår egen arbeidsflyt og praksis og forbedrer disse kontinuerlig.

Vi er ansvarlig for produktkvaliteten både under utvikling og i drift.





Prinsipp 5

Vi eksperimenter, måler og lærer raskt

Eksperimentering

Vi lager eksperimenter når vi er usikre, tester ulike hypoteser og analyserer resultatet. Vi tør å kaste det vi har investert i dersom det ikke gir verdi.

Måling

Vi søker innsikt via kontinuerlig måling av brukerbehov, produktkvalitet, prosesser, metoder og praksis, og følger med på trender over tid. Kvantitative målinger suppleres med kvalitative vurderinger og innsikt.

Læring

Vi lærer og gjør forbedringer kontinuerlig. Vi setter av tid til å lære av feil og deler lærdommen med andre.





Prinsipp 6

Vi er åpne og samarbeider med andre

Teamene deler det de produserer, for å tilrettelegge for samarbeid og læring.

Alle team har et ansvar for å ivareta helheten på tvers av produkter og team.

Samhandling og informasjonsdeling mellom involverte interessenter er åpen, tydelig og effektiv.





Prinsipp 7

Vi skaper tillit

Tillit innad i teamet og fra omgivelsene skaper trygghet til å ta gode beslutninger.

Vi stoler på at medlemmene i teamet:

- kan og vil ta ansvar for det som lages
- er i stand til å ta beslutninger til det beste for brukerne, utviklingen av produktet og NAV
- klarer å organisere seg og finne arbeidsprosesser som fungerer for dem
- finner enkle løsninger på komplekse utfordringer

Teamene definerer og prioriterer oppgaver som gradvis realiserer målbildet.



