

Kontinuerlig samskaping?^F

En casestudie av digitaliseringsinitiativer på Navs hjelpemiddelområde

SAMMENDRAG

© 2024 Astri Barbala, Tor Sporsem



Denne artikkelen tar for seg hvordan programvareutviklere på hjelpemiddelområdet i Nav samarbeider med brukere innad i organisasjonen om å utvikle nye, digitale saksbehandlingsløsninger med mål om å forbedre søknadsprosedyrene for fremtiden. For å undersøke dette har vi observert og intervjuet saksbehandlere på fire hjelpemiddelsentraler plassert på ulike steder i Norge for å forstå hvordan de tilpasser seg hverdagen med økende digitalisering. Vi har også observert og intervjuet et utviklerteam som lager saksbehandlingsprogramvare til denne brukergruppa. Studien fokuserer på hvordan utviklerne kommuniserer med saksbehandlerne i denne prosessen, der kontinuerlig programvareutviklings-

metodikk preger samskapingsarbeidet. Vi bruker Pierre Bourdieus (1986) begrep *sosial kapital* som analytisk verktøy og finner at det tverrfaglige samarbeidet mellom saksbehandlere og utviklere styrker Navs kollektive problemløsnings- evne. Samtidig er denne samskapingen også en verdifull investering som blir en forutsetning for at organisasjonen skal kunne høste fruktene av sine digitaliseringsinitiativer. Studien peker på at det tverrorganisatoriske samarbeidet muliggjør kontinuerlig tilpassning av programvaren slik at den enklere kan møte dynamiske brukerbehov. Likevel finnes det flere utfordringer i samskapingsarbeidet knyttet til å sikre nødvendig tilbakemelding fra brukerne.

Barbala, A., & Sporsem, T. Kontinuerlig samskaping? En casestudie av digitaliseringsinitiativer på NAVs hjelpemiddelområde. *Magma*, 27(2). <https://doi.org/10.23865/magma.v27.1446>

Korresponderende forfatter: Astri.Barbala@sintef.no



Astri Barbala jobber som forsker på Sintef i gruppa Digital Process Innovation og har en doktorgrad i sosiologi. Hennes forskningsinteresser er knyttet til hvordan digitalisering påvirker organisasjoner og mellommenneskelige forhold. Astri forsker for tiden på temaer som ansvarlig digital transformasjon og bruk av KI i offentlig sektor.



Tor Sporse forsker innen programvareutvikling i Sintef og er stipendiat ved NTNU. Tor er del av en forskningsgruppe i Sintef som har forsket på smidige (agile) metoder i over 20 år, og fokuserer på metoder som programvareutviklere bruker for å forstå brukerbehov.

Innledning

Norsk offentlig sektor har det siste tiåret blitt mer og mer digitalisert, og hverdagen til ansatte i offentlige organisasjoner blir stadig mer dominert av digitale verktøy i både behandling av saker og i sin kommunikasjon med brukere – de norske borgerne. Nav, en av de største offentlige etatene i Norge, er blant organisasjonene som har opplevd en digital omveltning de seneste årene. Mellom 2016 og 2020 gikk Nav gjennom et såkalt IT-løft som innebar at det ble investert mye ressurser i teknologiutvikling. I løpet av perioden ble det bygd opp et omfattende kompetansemiljø innen utvikling, data og design, og antallet interne utviklere økte med nærmere 250 personer mellom 2017 og 2019 (se Bernhardt, 2021). En viktig del av denne transformasjonen var Navs IT-avdeling, Nav IT, sitt strategiske grep med å innkontraktere IT-tjenestene sine, som resulterte i at programvareutviklere har blitt en mer integrert del av organisasjonen (Thunes & Kempton, 2023). Det har tilrettelagt for å kunne innføre en smidig og kontinuerlig arbeidsmetodikk. IT-bransjen har en stor fordel fremfor andre bransjer – de lager ikke et fysisk produkt. Det betyr at de kan gjøre store endringer på produktet sitt raskt og distribuere det til tusenvis av brukere på bare sekunder. Dette gjør at programvareutviklere kan eksperimentere kontinuerlig sammen med brukerne for å finne ut hva de ønsker seg. Til sammenligning må man i byggebransjen bygge huset helt ferdig før brukeren kan ta det i bruk og teste det. Gjennom innkontraktingen har Nav tatt tilbake kontrollen over denne prosessen, der innleide konsulenter før var i flertall. Videre har dette tilrettelagt for tverrfaglig samarbeid innad i organisasjonen som blant annet har betydd at utviklere har jobbet tett med jurister for mer effektivt å kunne iverksette datadrevne initiativer (Barbala et al., 2023).

I denne artikkelen tar vi for oss hvordan et slikt digitaliseringssamarbeid utspiller seg mellom utviklere og saksbehandlere på hjelpemiddelområdet i Nav. Denne delen av organisasjonen forventer en enorm økning i borgere som trenger hjelpemidler som rullestoler, høreapparater

og sitteputer i årene som kommer. Ved å digitalisere deler av prosessen skal hver saksbehandler kunne hjelpe flere borgere og behandle flere saker på kortere tid, noe som blir helt nødvendig når antall saker stiger. Denne studien ser på digitaliseringsarbeidet på hjelpemiddelområdet i form av kontinuerlig programvareutvikling. Per dags dato har få studier undersøkt hvordan brukere involveres i slik utvikling, og særlig lite forskning har sett på hvordan dette utspiller seg i en organisasjonssetting eller innen offentlig sektor. De offentlige etatene er i en unik situasjon, da programvareutviklingen og brukerinvolvering her ikke drives av et økonomisk insentiv, men mer dreier seg om at samarbeid på tvers av organisasjonen er nødvendig for å kunne oppfylle samfunnsoppdraget som er en del av det å være offentlig ansatt. Vi spør: Hvilke hindre finnes for at samarbeidet mellom programvareutviklere og ekspertbrukere innad i Nav skal fungere optimalt i arbeidet med å utvikle nye, digitale saksbehandlingsløsninger?

Begge parter er avhengig av tett samarbeid med den andre for å både forbedre sin arbeidshverdag og få fart på Navs digitaliseringsarbeid.

For å finne svar på dette har vi observert og intervjuet saksbehandlere på fire hjelpemiddelsentraler på ulike steder i Norge. Vi har også intervjuet et utviklerteam som lager saksbehandlingsprogramvare på hjelpemiddelområdet, og vi har fulgt dem i utviklingsprosessen for å se hvordan de jobber sammen med saksbehandlerne for å effektivisere og forenkle saksbehandlingsprosessen. Artikkelen bygger på 19 intervjuer (åtte fra utviklerteamet og elleve fra hjelpemiddelsentralene) og fem dager med observasjoner som ble utført både hos utviklerteamet og på de fire hjelpemiddelsentralene. I tillegg har vi hatt flere møter og samtaler med sentrale interessenter på hjelpemiddelområdet. Vi bruker begrepet sosial kapital (Bourdieu, 1986) som teoretisk og analytisk rammeverk for å diskutere funnene våre. Vi knytter sosial kapital sammen med samskapingsbegrepet for å vise hvordan det å utvikle sosiale relasjoner og nettverk representerer ressurser som kan styrke den kollektive problemløsningsevnen innad i organisasjoner som Nav.

Denne artikkelen er en videreutvikling av artikkelen «Continuous adoption in the Norwegian public sector» (Barbala et al., 2024), der fokuset er å videreutvikle rammeverket for kontinuerlig programvareutvikling til også å omfatte hvordan programvaren blir en del av hverdagen til brukerne. Selv om vi bygger på det samme datamaterialet, er dog vårt fokus i denne artikkelen å se på hvordan

programvareutviklere samarbeider med ekspertbrukere for å skape verdi og avkastning i Navs digitaliseringsarbeid.

Kontinuerlig programvareutvikling

En viktig tekst innen kontinuerlig programvareutviklings-teori er Brian Fitzgerald og Klaas-Jan Stols «Continuous software engineering: A roadmap and agenda» (2014). Forfatterne argumenterer her for at begrepet *flow* fra Lean Thinking-metodikken kan understreke det «sammenkoblede settet av verdiskapende handlinger» (Fitzgerald & Stol, 2014; 180) som ligger til grunn for kontinuerlig programvareutvikling. De hevder at vekt på ende-til-ende-prosesser kan være bedre egnet for den kontinuerlige metodikken enn smidig programvareutvikling, som er mindre opptatt av *hele kjeden* av integrerte praksiser. Et sentralt prinsipp i den kontinuerlige tilnærmingen er å ha en pågående relasjon til brukerne underveis i hele prosessen for best å kunne forstå deres behov og utfordringer.

Et sentralt prinsipp i den kontinuerlige tilnærmingen er å ha en pågående relasjon til brukerne underveis i hele prosessen for best å kunne forstå deres behov og utfordringer.

Å etablere tillit til brukere er sammenvevd med implementering av brukertilbakemeldinger og er derfor en viktig del av denne prosessen. Klotins et al. (2022) har sett på kontinuerlig utvikling i deler av privat sektor, der brukere som regel er kunder som har betalt for å bruke programvaren, og skriver: «De primære fordelene ved kontinuerlig [metodikk] kommer fra tettere relasjoner med kunder og sluttbrukere. Denne relasjonen gir flere muligheter for innsamling av tilbakemeldinger, det bygger tillit [mellom utviklere og brukere] og forbedrer den generelle kundetilfredsheten.» Det er dog grunn til å tro at forholdet mellom brukere og utviklere utspiller seg på en annen måte når brukerne ikke er kunder. Denne artikkelen vil bidra med verdifull innsikt om hvordan tillitsbyggingen utspiller seg når det ikke er noe økonomisk insentiv for å inkludere brukerne i utviklingsprosessen, og spesielt hvordan dette foregår i en organisasjonssetting.

Sosial kapital: Relasjoner som ressurs

Den franske sosiologen og antropologen Pierre Bourdieu utformet teorien om sosial kapital som et nøkkelbegrep i

sitt omfattende arbeid med å utvide forståelsen av kapital til også å omhandle ikke-økonomisk kapital (Bourdieu, 1986). Bourdieu introduserte begrepet for å beskrive de sosiale forbindelsene som en person har, herunder relasjonene til andre individer, organisasjoner og institusjoner. En person eller et nettverk med stor sosial kapital kan hente ut avkastning i form av hjelp og personlige tjenester gjennom relasjonene sine. Sentralt i Bourdieus tenkning er oppfatningen om at sosial kapital utgjør et viktig element for å forstå samarbeidsmulighetene og dynamikken mellom ulike aktører. Sosial kapital gir oss ifølge Bourdieu altså tilgang til ressurser gjennom nettverksbygging og samarbeid.

Selv om Bourdieu først og fremst så på sosial kapital som en egenskap hos individet snarere enn kollektivet, har organisasjonsforskere nyttiggjort seg av begrepet for å diskutere samhandlingsprosesser innad i organisasjoner. For eksempel har en mye sitert studie av Jane Dutton og Belle Ragins (2017) sett på hvordan positiv sosial kapital skapes og opprettholdes i organisasjoner. De avdekker at de underliggende mekanismene for sosial kapital som gagnar både organisasjonen og enkeltindivider, er *motivasjon* og *mulighetsstrukturer*, som gunstige kommunikasjonskanaler og nok ressurser. Videre peker Dutton og Ragins på viktigheten av å fokusere både på *midlene* som sosial kapital er skapt med, og *målene* den brukes til, for at sosial kapital skal kunne kobles til tverrorganisatorisk samskaping: «Sosial kapital er positiv hvis midlene den er skapt med utvider kapasiteten til både enkeltmennesker og grupper. Sosial kapital er positiv hvis den hjelper mennesker til å utvikle seg, trives og blomstre i organisasjoner og dermed oppnå målene sine på bedre måter» (2017, s. 2, oversatt fra engelsk).

Innen forskning på programvareutvikling har også Bourdieus begrep funnet fotfeste, og blant annet har Moe et al. (2014) brukt sosial kapital-begrepet for å studere kunnskapsressursene som er bygd inn i utviklerteamenes nettverk, der hvem du kjenner, påvirker kunnskapen din direkte. Tilsvarende har Wohlin et al. (2015) sett på viktigheten av sosial kapital både innad i et team og mellom teamet og eksterne samarbeidspartnere som brukere eller kolleger i andre deler av organisasjonen, der tillit til personer utenfor ens eget team og identifikasjon med programvaren er viktige stikkord. De utformer følgende definisjon på sosial kapital i en programvareutviklingskontekst: «De faktiske og potensielle ressursene som er innebygd i, tilgjengelige gjennom og kan utledes av relasjonsnettverket til en individuell aktør eller en sosial gruppe.» Vi anser denne definisjonen som gunstig også for vår studie, da den peker både på at utviklerteamet og saksbehandlerne stiller med sine respektive ressurser, og på at disse sammenlagt skaper verdi, i form av individuell kunnskaps- og relasjonsbygging, bedre programvareutvikling og til syvende og sist også økt avkastning for Nav og samfunnet for øvrig. Ved å anvende teorien om sosial kapital i denne artikkelen søker vi å skape innsikt i hvordan sosiale nettverk kan være avgjørende for å forstå og forbedre tverrfaglige programvarearbeidsprosesser også på tvers av komplekse, offentlige sektororganisasjoner.

Metode og datamateriale

For å belyse problemstillingen best mulig gjennomførte vi en utforskende casestudie inspirert av tilnærmingen til Per Runeson og Martin Höst (2008). De skisserer fem trinn for å utføre denne typen studie innen forskning på programvareutvikling: (1) studiedesign, (2) forberedelse til datainnsamling, (3) innsamling av data, (4) analyse av innsamlede data og (5) rapportering. Datamaterialet består av en kombinasjon av observasjoner av og intervjuer med et utviklerteam på hjelpemiddelområdet og saksbehandlere fra fire ulike hjelpemiddelsentraler som bruker programvaren utviklet av teamet. Både utviklerteamet og saksbehandlerne ble observert i en naturlig setting i løpet av sin vanlige arbeidshverdag. Videre omfattet datainnsamlingen dokumentstudier av strategidokumenter og rapporter som beskriver status for digitaliseringsinitiativene i Nav.

For å sikre et helhetlig og representativt utvalg plukket vi ut fire hjelpemiddelsentraler lokalisert i ulike deler av landet, og de utvalgte sentralene dekker både store og små kommuner. Saksbehandlerne vi intervjuet og observerte, kan karakteriseres som *ekspertbrukere* av programvaren som er utviklet for dem, og den digitale saksbehandlingsløsningen vi fokuserte på, er skreddersydd for å møte disse brukernes behov i sin spesifikke arbeidssituasjon. Løsningen er tilpasset saksbehandlere på tvers av 15 hjelpemiddelsentraler i ulike kommuner i hele Norge som daglig behandler søknader og bestillinger fra folk som har behov for ulike hjelpemidler.

Utviklerteamet vi intervjuet, syntes det var vanskelig å avgjøre hvor stor en endring i programvaren måtte være før det var nødvendig å informere brukerne.

Selv om programvaren brukes av mellom 100 og 200 saksbehandlere på sentralene, er det viktig å merke seg at disse saksbehandlerne har ansvar for ulike hjelpemiddelområder. Det betyr at ikke alle bruker de samme digitale verktøyene eller behandler den samme typen digitale søknader. Intervjuene vi utførte, fulgte et semistrukturert format og varte mellom 30 og 60 minutter hver. Totalt gjennomførte vi 18 intervjuer med 19 informanter. Intervjuene fokuserte først og fremst på saksbehandlerne bruk av programvaren, deres erfaringer med å delta i utviklingsprosessen og hvordan disse aspektene ble oppfattet både fra utviklings- og brukerperspektivet.

Datainnsamlingen vår omfattet omtrent 30 timer med observasjoner på hjelpemiddelsentralene, med rundt sju

timer på hver sentral og fire timer med observasjoner av utviklerteamet. Vi deltok også i to Teams-møter og analyserte samtaler i Teams-chatterom. Intervjuene ble transkribert og kodet ved hjelp av NVivo for å identifisere og strukturere kodene til overordnede temaer. Ett sentralt tema som dukket opp i løpet av dataanalysen, var «tilbakemeldinger fra brukere», der undertemaene blant annet var «tverrfaglig forståelse», «juridiske utfordringer» og «ulike typer kommunikasjon» (f.eks. Teams-møter og kommunikasjon ansikt til ansikt). Begrepet sosial kapital ble ansett som et passende analytisk verktøy for å analysere disse temaene, som vi vil gå nærmere inn på i det følgende.

Funn: To fundamentale utfordringer i å sikre tilbakemeldinger

Gjennom dataanalysen fant vi at samarbeidet med å bygge nye, digitale løsninger på hjelpemiddelområdet i Nav avhenger mye av tilbakemeldinger fra saksbehandlerne til utviklerteamet, og at hindrene for sosial kapital hovedsakelig finnes her. I denne delen vil vi derfor vise to eksempler på dette.

Ett av kjerneargumentene for å drive med kontinuerlig programvareutvikling, altså at man utvikler programvaren bit for bit, er at utviklerne kan få rask tilbakemelding fra brukerne på det de lager. Slik kan de hele tiden justere arbeidet sitt etter hva brukerne mener er verdifullt. Våre funn avdekket imidlertid to grunnleggende utfordringer med denne metoden på hjelpemiddelområdet som kan demotivere brukere fra å gi tilbakemeldinger: (1) Hvordan skal man holde brukerne informert om *hva* som skal lages, og *når* det lages, når det skjer endringer konstant? (2) Det å formulere selve tilbakemeldingene skaper usikkerhet hos saksbehandlerne. Vi vil hevde at begge disse punktene henger sammen med at det ikke er opprettet tette nok relasjoner mellom brukerne og utviklerne, slik at de to gruppene verken kjenner til hverandres hverdag og arbeidsoppgaver godt nok eller føler seg trygge nok til å ta kontakt når de har behov for å formidle noe.

1) Derfor er det vanskelig å holde brukere informert

I tradisjonell prosjektmetodikk er det vanlig med milepæler og tidsfrister for leveranser. Selv om man sjelden når disse tidsfristene, skaper det en følelse av forutsigbarhet hos brukerne. Det bidrar til at brukerne forstår hvor utviklerne er i prosessen med å lage ny programvare til dem. I moderne metoder for programvareutvikling benytter man seg ikke av tidsfrister fordi forskning (se f.eks. Fitzgerald & Stol, 2014) har pekt på at korte planleggingshorisoner og kontinuerlig omprioritering av hva som skal utvikles, skaper bedre resultater. Utfordringen er da at det ikke er noen naturlige tidspunkter for å informere om nye funksjoner eller endringer. Utviklerteamet vi intervjuet, syntes det var vanskelig å avgjøre hvor stor en endring i programvaren måtte være før det var nødvendig å informere brukerne. Dette kan føre til at utviklerne ikke informerer om hva de jobber med, for å skåne saksbehandlerne. Effekten blir at saksbehandlerne ikke får innblikk i hva som skjer, og kan føle seg oversett.

En annen utfordring handler om å bestemme *når* det er riktig å gi informasjon. I kontinuerlig utvikling er

det konstant søkelys på hva som er verdt å utvikle, og det omprioriteres ofte for å sikre at det som lages, er verdifullt for brukerne, noe som betyr at utviklere aldri er helt sikre på når nye funksjoner blir lansert (eller om de i det hele tatt blir gjort ferdig). Intervjuene våre avdekket at saksbehandlere ofte ønsker å få detaljert informasjon om når en ny funksjon lanseres, slik at de kan forberede seg. Forberedelsene kan for eksempel være å øke bemanningen mens de tilpasser seg endringen, eller å sette av tid til å lære hvordan endringen påvirker deres arbeidsmetoder. Som et eksempel hadde en sentral kun én person som håndterte sakene i det nye systemet, mens resten av teamet var «beskyttet» fra dette arbeidet og kunne gjøre sine vanlige oppgaver. En saksbehandler med lederansvar forklarte det slik:

Det er greit å vite for eksempel, 'om to uker kobler vi på rammeavtaler for løftestoler [i det nye, digitale systemet]'. Da vet vi at vi har noe å forholde oss til. Vi kan forberede oss litt, vi kan vurdere om det er de samme folkene som skal fortsette [å ha ansvaret for nye, digitale løsninger], eller om må vi sette på noen som har mer kompetanse på akkurat det produktområdet på de søkene.

Med andre ord: Hvis flere oppgaver overføres til det nye systemet, må sentralen omorganisere ansvarsområdene, og dette krever forberedelse. Dette skaper et dilemma for utviklerteamet: Hvis man informerer *for sent*, har ikke saksbehandlere tid til å forberede seg. Men hvis man informerer *for tidlig*, kan saksbehandlere ende med å forberede seg på noe som aldri kommer.

Saksbehandlerne var ofte usikre på om det var en feil i saksbehandlings-systemet, eller om det var brukerfeil fra deres side.

Å informere om at det kommer en ny funksjon som viser seg å ikke komme likevel, kan bryte ned tilliten hos brukerne. Slik kan metodene for kontinuerlig utvikling skape et vanskelig dilemma for utviklerteamene, som risikerer å svekke brukernes tillit til dem, noe som igjen er et hinder på veien for å skape nødvendig sosial kapital.

Brukerne føler seg også lite inkludert når de ikke forstår hva som prioriteres og hvorfor. «Det hadde vært fint med en mer komplett oversikt på hva man har fått tilbakemeldinger på, hvordan de har vurdert det, og i hvilken grad de prioriterer det», sa en saksbehandler, og la til: «Det er ikke noen motivasjon i seg selv å gi tilbakemeldinger når du ikke vet hva som er gjort med tilbakemeldingene.» Tilbakemeldin-

ger som handler om at det er en feil i programvaren, altså en *bug* i systemet, blir raskt tatt tak i og fikset kanskje allerede samme dag. Tilbakemeldinger som handler om endringsforslag, tar dog lang tid å behandle fordi det må vurderes om endringen er viktigere enn annet utviklingsarbeid som foregår. Brukerne kan dermed ende med å føle at tilbakemeldingen deres aldri tas tak i.

Det er viktig å nevne at det å holde styr på hvilke brukere som har kommet med hvilke tilbakemeldinger, og informere dem om at tilbakemeldingene har blitt implementert, er en mer komplisert oppgave enn det kan virke som. Én tilbakemelding kan føre til mange små oppgaver for utviklerne, og disse kan endre seg etter hvert som arbeidet gjøres. Når ting stadig blir omprioritert og endret, er det vanskelig å følge med på hvilke resultater som kan kobles til hvilke forslag. I tillegg kan flere brukere foreslå det samme på forskjellige tidspunkter, noe som gjør det enda mer komplisert å holde oversikt og gi tilbakemelding. Om det er verdt for utviklerne å bruke tid på å lage disse oversiktene, eller om de heller bør bruke denne tiden på å utvikle programvare, er vanskelig å si sikkert. Dette illustrerer en av de store utfordringene utviklerteamet står i.

2) Det er utfordrende for brukere å gi tilbakemeldinger

Funnene våre viser også at utviklere risikerer å gå glipp av viktige tilbakemeldinger hvis brukerne er usikre på hvordan de skal gi dem. I vår studie kunne brukerne sende inn tilbakemeldinger gjennom spesifikke Teams-kanaler, noe flere saksbehandlere syntes var ubehagelig. Årsaken var at de ofte var usikre på om det var en feil i saksbehandlingssystemet, eller om det var brukerfeil fra deres side. De risikerer å dumme seg ut foran kolleger de ikke kjenner, når de melder fra i en åpen kanal. «Du har jo et publikum når du skriver på en vegg [kanal]», uttrykte en saksbehandler.

Flere forteller at de heller tar direkte kontakt med noen i utviklerteamet når de kjenner på denne usikkerheten. Da tør de stille «dumme» spørsmål og forsikre seg om at de har forstått ting riktig. Men dette ser ut til å kun skje blant de saksbehandlerne som har en etablert relasjon til noen i utviklerteamet. Det er kjent fra tidligere forskning at det er mye lavere terskel for å ta kontakt digitalt med folk man allerede kjenner, fordi normer i relasjonen da er etablert (Sporssem & Moe, 2022). Dette viser viktigheten av en personlig relasjon mellom utviklerteamet og brukere for å sikre seg nødvendig tilbakemelding – noe som kanskje er ekstra viktig når det er så få brukere, altså få muligheter til å få verdifulle innspill. Et resultat av dette er at utviklerteamet risikerer å gå glipp av tilbakemeldinger fra sentraler som det ikke er etablert en tydelig relasjon til. Et medlem av utviklerteamet uttrykte viktigheten av å ha faste Teams-møter med saksbehandlerne for å kunne etablere disse relasjonene som fører til at tilbakemeldingene kommer frem:

Det er det mest nyttige vi gjør, og det er kjempeverdifullt. Og det er verdifullt fordi vi får innsikt i hvordan de bruker løsningene, men også fordi

vi får relasjoner til disse menneskene. Og da er det lettere for dem, og for oss, å snakke om andre ting også. [Saksbehandlerne] sier: 'Ja, når vi først har dere her, så ønsker vi oss disse og disse tingene.' Sannsynligvis er det ting de ellers ikke hadde gitt oss tilbakemeldinger på.

Vi ser også at saksbehandlerne holdning til utviklerne spiller en viktig rolle. Saksbehandlerne hadde stor respekt for utviklerne og anerkjente at de har en vanskelig og kompleks jobb, uten å ha inngående kjennskap til deres hverdag. Derfor ønsket de ikke å forstyrre utviklerne med små detaljer som de mente ikke var viktige. Det kan imidlertid virke som det saksbehandlerne anser som uviktige tilbakemeldinger, ofte er viktige for utviklerne. Ironisk nok kan denne respekten og ydmykheten overfor utviklerne virke mot deres ønske om å få så hyppige tilbakemeldinger som mulig, om flest mulig aspekter ved programvaren.

Om saksbehandlerne ikke får behandlet søknader på en tilfredsstillende måte, er det brukere av hjelpemidler i befolkningen som til syvende og sist blir teknologiske forsøkskaniner

Det er mellom 110 og 180 saksbehandlere som er brukere av saksbehandlingssystemet vi har studert. Dette faste antallet brukere fører med seg en stor risiko: Om de demotiveres og slutter å gi tilbakemeldinger, er det ikke mulig å rekruttere andre brukere. Sett i sammenheng med apper som lages for millionvis, er dette veldig få brukere. Saksbehandlerne kan ikke velge en konkurrerende løsning, slik man ofte kan med annen programvare der man kan velge leverandør i et marked. Holdningen blant saksbehandlerne er derfor at de må «gjøre det beste ut av det» siden utviklerteamet driver med avanserte oppgaver som de ikke forstår så mye av. Dette risikerer å skape en situasjon hvor brukerne må tilpasse seg utviklerne prosess, og de mister følelsen av å kunne påvirke eget arbeid. På en sentral vi besøkte, forklarte for eksempel flere at noen saksbehandlere har mistet motivasjonen til å gi tilbakemeldinger. «De hos oss som er mest kritiske til utviklingen, vil ikke ha kontakt med [utviklerne], noe som bekymrer meg», forteller en ansatt på en sentral. Det kan tyde på at viktige tilbakemeldinger ikke kommer frem, når brukerne har fått en negativ hold-

ning til utviklingsarbeidet. Manglende tid og ressurser ble ofte nevnt som hovedgrunnen til denne uvilligheten til å ha kontakt med utviklerteamet, men det kan også tenkes å være en følge av manglende sosiale relasjoner de to gruppene imellom.

Man risikerer å demotivere brukerne når man knebler følelsen av å kunne bestemme over egen arbeidsprosess, selv om dette ikke er utviklersidens mål – snarere tvert imot. Som et medlem av utviklerteamet forklarte: «[Jeg er opp-tatt av] at de skal føle seg trygge og føle at de kan snakke åpent med oss om ting, og ikke føle seg vurdert av oss. Prøve å skape det trygge rommet. Jeg er veldig opptatt av å ha en relasjon til disse folkene.» Samtidig uttrykte samme person frustrasjon over vanskene med å gjøre seg forstått grunnet ulike fagbakgrunner: «Jeg tenker jo ikke at de er dumme som ikke forstår det, jeg tenker jo at vi må gjøre det annerledes. Men jeg føler at jeg får litt sånn fattig språk etter hvert.» Mangelen på sosial kapital ser derfor her ut til å være forårsaket av fravær av tydelig kommunikasjon dem imellom: Begge parter er avhengig av tett samarbeid med den andre for å både forbedre sin arbeidshverdag og få fart på Navs digitaliseringsarbeid, som til syvende og sist gagnar hele organisasjonen. Dette er også i tråd med funnene i Dutton & Ragins' (2017) studie av tverrorganisatorisk samarbeid. Her er det imidlertid viktig å poengtere at hovedårsaken til dette ikke er at noen av gruppene svikter i sine forsøk på å prøve å etablere denne relasjonen, men snarere at det å opprette tett kontakt mellom brukere og utviklere i seg selv er ressurskrevende arbeid som kommer i tillegg andre presserende arbeidsoppgaver.

Konkluderende diskusjon

Basert på de nylig presenterte funnene vil vi nå diskutere hvordan prinsippene om kontinuerlig utvikling kan være til hinder for dannelsen av sosial kapital i en tverrorganisatorisk offentlig sektor.

Prinsippene som ligger til grunn for kontinuerlig programvareutvikling, der programvare utvikles og slippes til brukeren bit for bit (se f.eks. Fitzgerald & Stol, 2014), legger føringer både for hvilket fokus digitaliseringsinitiativene får, og hvordan samarbeidet om disse initiativene foregår på tvers av organisasjonen. Når det lages skreddersydd programvare til en liten brukergruppe som ikke har mulighet til å velge bort programvaren til fordel for en konkurrerende leverandør, stiller det høye krav til brukerinvolvering og -forståelse. Et av kjerneargumentene for å drive med kontinuerlig utvikling er at man får testet det man lager, sammen med brukerne. Slik får man tilbakemeldinger underveis og passer på at man lager noe som brukerne faktisk trenger. Men om disse tilbakemeldingene slutter å komme, forsvinner også den store fordelene med kontinuerlig programvareutvikling. Vi finner at dette er et fundamentalt problem når utviklerne er låst til så få brukere. Dette er i liten grad anerkjent i litteraturen om smidig og kontinuerlig utvikling fordi metodene er studert i bedrifter med titusenvis til millionvis av brukere som man kan plukke tilbakemeldinger fra. Som vi har pekt på, gjør dette at utviklerne i denne situasjonen må tenke annerledes rundt det å opprette tillit til brukerne sine (Klotins et al., 2022).

En mye sitert studie av Maalej et al. (2009) som tar for seg hvordan brukere i kontinuerlig programvareutvikling blir samarbeidspartnere i utviklingsarbeidet, understreker viktigheten av at utviklerne kjenner brukernes spesifikke kontekst. Dette er kanskje særlig viktig når programvaren skreddersys til en liten ekspertgruppe som bruker den i sitt daglige virke. Å holde dette antallet brukere motivert til å gi tilbakemeldinger er derfor essensielt. Noe som var tydelig i våre funn, var at tidshorizonten som ligger innbakt som en sentral del av den kontinuerlige metodikken, ikke nødvendigvis er kompatibel med å informere brukerne nok om den teknologiske siden ved digitaliseringsarbeidet. Ekspertbrukere som saksbehandlere har ikke mulighet til å konstant eksperimentere med utviklerne eller «teste i produksjon» til stadighet underveis, fordi det står for mye på spill: Om de ikke får behandlet søknader og bestillinger på en tilfredsstillende måte, er det brukere av hjelpemidler i befolkningen som til syvende og sist blir teknologiske forsøkskaniner og risikerer å ikke få den hjelpen de trenger.

Som vår studie illustrerer, er dette imidlertid ikke noe som står i fare for å skje hos Nav. Selv om det var gjensidig tillit og respekt mellom de to ulike gruppene, var likevel for lite tid til å skape relasjoner en utfordring. For å sikre reell forståelse seg imellom er saksbehandlerrollen og utviklerrollen nødt til å nærme seg hverandre både faglig og sosialt for best mulig å kunne skape grobunn for sosial kapital, ved å utvikle et tverrfaglig språk som kan brukes i utviklingsprosessen. Dette er dog en ressurskrevende jobb, og vi fikk inntrykk av det ikke var ønsket om dette som var avgjørende, men snarere at både utviklingen av nye, digitale løsninger og saksbehandling av hjelpemiddelsøknader er ytterst krevende jobber med mange ulike prioriteringer. Vi så også at det er summen av de små aktivitetene, gjennom hyppige samtaler på for eksempel Teams eller faste, ukentlige møter, som til sammen kan skape verdifull sosial kapital. Et annet eksempel var der en saksbehandler hadde fått særlig god kontakt med en av utviklerne. Ikke bare gjorde dette at tilbakemeldinger og beskjeder fløt bedre begge veier – det gjorde også at disse personene var mer positivt innstilt til samskapsarbeidet med å utvikle nye digitale løsninger. Dette funnet samsvarer også med tidligere forskning på sosial kapital i organisasjoner (se Dutton & Ragins, 2017).

Selv om vår studie har bidratt med ny kunnskap omkring hvordan kontinuerlig programvareutvikling og tverrfaglige digitaliseringsinitiativer utspiller seg i praksis i offentlig sektor, trengs det fremdeles mer forskning som kan være med på å belyse de ulike digitaliseringsutfordringene samfunnet står ovenfor. På hjelpemiddelområdet er det særlig den kommende eldrebølgen som utpeker seg, og fremtidig forskning bør derfor også studere hvordan utviklere i offentlig sektor kan favne eldre borgeres brukerinnspill i sitt utviklingsarbeid, da dette vil være en dominerende brukergruppe i fremtiden.

Vi vil takke Nav og våre informanter for å ha delt sine erfaringer med oss, samt Knowit AS og Norges forskningsråd, som har finansiert forskningen (prosjektnummer 321477).

REFERANSER

- Barbala, A., Sporsem, T. & Stol, K. J. (2024). A case study of continuous adoption in the Norwegian public sector. *Hawaii International Conference on System Sciences (HICSS 2024, Honolulu)*.
- Barbala, A., Sporsem, T. & Stray, V. (2023). Data-driven development in public sector: How agile product teams maneuver data privacy regulations. I C. J. Stettina, J. Garbajosa & P. Kruchten (Red.), *Agile processes in software engineering and extreme programming*, 475 (s. 165–180). Springer International Publishing.
- Bernhardt, H. B. (2021). *Endringsreisen i NAV IT 2016–2020: En studie av den digitale transformasjonen i NAV IT og endringenes effekt på håndtering av koronakrisen* [Masteroppgave]. Norges teknisk-naturvitenskapelige universitet.
- Bourdieu, P. (1986). The forms of capital. I J. G. Richardson (Red.), *Handbook of theory and research for the sociology of education* (s. 241–258). Greenwood Press.
- Dutton, J. & Ragins, B. (2017). *Exploring positive relationships at work: Building a theoretical and research foundation*. Taylor & Francis.
- Fitzgerald, B. & Stol, K.J. (2014). Continuous software engineering and beyond: Trends and challenges. *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering* (s. 1–9).
- Klotins, E., Gorschek, T., Sundelin, K. & Falk, E. (2022). Towards cost-benefit evaluation for continuous software engineering activities. *Empirical Software Engineering*, 27(6), 157.
- Maalej, W., Happel, H. J. & Rashid, A. (2009). When users become collaborators: Towards continuous and context-aware user input. *Proceeding of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, OOPSLA 2009* (s. 981–990).
- Moe, N. B., Šmite, D., Hanssen, G. K. & Barney, H. (2014). From offshore outsourcing to insourcing and partnerships: Four failed outsourcing attempts. *Empirical Software Engineering*, 19, 1225–1258.
- Runeson, P. & Höst, M. (2008) Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131–164.
- Sporsem, T. & Moe, N. B. (2022). Coordination strategies when working from anywhere: A case study of two agile teams. *International conference on agile software development* (s. 52–61). Springer International Publishing.
- Thunes, J. & Kempton, A. M. (2023). Achieving digital transformation in the public sector through targeted insourcing. *ECIS 2023 Research Papers*.
- Wohlin, C., Šmite, D. & Moe, N. B. (2015). A general theory of software engineering: Balancing human, social and organizational capitals. *Journal of Systems and Software*, 109, 229–242.