

SOA-prosjekter i praksis:

Erfaringer fra en casestudie i Skatteetaten

Christer Eikrem de Lange

Veileder

Eli Hustad

Masteroppgaven er gjennomført som ledd i utdanningen ved Universitetet i Agder og er godkjent som del av denne utdanningen. Denne godkjenningen innebærer ikke at universitetet inntår for de metoder som er anvendt og de konklusjoner som er trukket.

Forord

Denne studien ble skrevet som en avsluttende oppgave ved Mastergradsstudiet i Informasjonssystemer ved Universitetet i Agder, høsten 2012.

Studien har hatt til hensikt å følge et SOA-prosjekt i praksis for å øke kunnskapen om hvilke faktorer som påvirker systemutvikling i et slikt prosjekt. Studien ble gjennomført som en singel casestudie basert på et fortolkende perspektiv. Jeg fulgte «Dokumentprosjektet» over ett år og intervjuet sentrale deltakere underveis. Studien bygger på et teoretisk grunnlag basert på anerkjente rammeverk og relevant litteratur innen tjenesteorientert arkitektur og smidig systemutvikling.

Veileder for studien har vært Eli Hustad ved Universitetet i Agder. Jeg vil gjerne takke henne for meget god veiledning gjennom studien, og for alle innspill og tilbakemeldinger underveis. Jeg vil også rette en stor takk til min kontaktperson i Skatteetaten, Lene Staverløkk, som har tatt seg tid til å svare på spørsmål og forklare hvordan Skatteetatens mange prosjekter henger sammen. I tillegg vil jeg takke alle deltakerne i studien for at de tok seg tid til å stille opp på intervjuer og svare på spørsmål i forbindelse med studien.

Til sist vil jeg rette en stor takk til min kone og min sønn for stor tålmodighet og støtte under arbeidet med masteroppgaven.

Kristiansand, 11.12.2012

Christer Eikrem de Lange

Sammendrag

Denne masteroppgaven bygger på en studie i Skatteetaten. Fokus i studien har vært å følge systemutviklingsprosjekter ved innføringen av felleskomponenter i en tjenesteorientert arkitektur (Service-oriented architecture, SOA).

Studiens målsetning har vært å øke kunnskapen om hvilke faktorer som påvirker systemutvikling i en kompleks tjenesteorientert kontekst. Dette ble gjort ved å følge innføringen av en fellestjeneste og tilhørende brukertjenester ved utvikling av en felles dokumentløsning i Skatteetaten. Ved å benytte et sosio-teknisk perspektiv har studien vært fokusert på å besvare følgende forskningsspørsmål:

1. Hvordan gjennomføres SOA-prosjekter i praksis?
2. Hvilke utfordringer er sentrale ved innføring av fellestjenester i en tjenesteorientert arkitektur?
3. Hvordan kan en organisasjon ta lærdom av denne type prosjekter?

For å besvare dette har studien bygget på et teoretisk grunnlag rundt anerkjente rammeverk og relevant litteratur som fokuserer på innføring av tjenesteorientert arkitektur og styring av SOA-prosjekter. Studien tar også for seg smidig systemutvikling og har sett på hvordan dette påvirker systemutviklingsprosjekter som er tjenesteorienterte i fokus.

For å oppnå studiens målsetning ble det gjennomført en kvalitativ utforskende casestudie basert på et fortolkende perspektiv. Skatteetaten startet sitt langsiktige SOA-program i 2009, og flere prosjekter har vært gjennomført. Denne masteroppgaven bygger på Skatteetatens SOA-prosjekt «Dokumentprosjektet» som ble fulgt over ett år. Den primære datakilden har bestått av semistrukturerte intervjuer. Intervjuene ble gjennomført ved ulike tidspunkter i løpet av 2012, og deltakerne i studien var prosjektledere, arkitekter og systemutviklere. Deltakerne ble valgt strategisk for å kunne sikre et rikt datagrunnlag.

Studien bidrar til å øke forståelsen for hvordan SOA-prosjekter gjennomføres i praksis.

Dokumentprosjektet utgjør et viktig signalprosjekt for Skatteetaten, og erfaringer fra prosjektet bør tas videre inn i nye prosjekter. Studien har identifisert utfordringer relatert til gjennomføring av SOA-prosjekter. Utfordringene er delt inn i seks hovedkategorier: 1) Kompetanse 2) Koordinering 3) Kommunikasjon 4) Parallele prosjekter 5) Konsulentbruk 6) Prosjekt- og systemutviklingsmetodikk.

Studien ble gjennomført i en spennende og utfordrende fase - overgangen fra et overordnet SOA-program til innføringen av konkrete fellestjenester i praksis. Funn viser imidlertid at man må høste nødvendige erfaringer fra IT-arkitekturen underveis før man starter nye systemutviklingsprosjekter som videreutvikler IT-arkitekturen. Flere utfordringer oppstod fordi man på tidspunktet for igangsetting av prosjektene ikke hadde tilstrekkelig kunnskap og erfaringer relatert til hvordan SOA-prosjekter skulle styres i praksis (SOA governance). Studien konkluderer med at det er en forskjell mellom å gjennomføre et systemutviklingsprosjekt innad i en silo og det å utvikle fellestjenester og fagkomponenter på tvers av organisasjonen. Prosjektledere som innfører fellestjenester bør fokusere på kompleksitet i utviklingskonteksten og ha et helhetlig perspektiv på tjenestene. For å lykkes med innføring av fellestjenester i en tjenesteorientert arkitektur er det viktig å fokusere på dette og de seks hovedkategoriene som studien identifiserer.

Innholdsfortegnelse

1. Innledning.....	1
1.1 Motivasjon.....	1
1.2 Problemstilling og forskningsspørsmål.....	2
1.3 Leseveiledning.....	2
2. Relevant tidligere forskning.....	3
2.1 Hva er SOA?.....	3
2.1.1 Hva er en tjeneste (service)?.....	4
2.1.2 Hva er en prosess?.....	5
2.1.3 Tjenesteorientert arkitektur.....	6
2.2 IT-arkitektur.....	7
2.3 Kompleksitet og sosio-teknisk perspektiv.....	10
2.4 Rammeverk for implementasjon av SOA.....	12
2.4.1 BEA domain model.....	13
2.4.2 Oracle SOA Maturity Model.....	15
2.4.3 SOA governance.....	17
2.4.4 Oppsummering rammeverk.....	20
3. Smidig systemutvikling.....	21
3.1 Smidige metoder.....	21
3.2 Scrum.....	23
4. Casebeskrivelse av Skatteetaten.....	25
4.1 Skatteetatens organisasjon.....	25
4.2 SOA i Skatteetaten.....	26
4.3 Tidligere prosjekter i Skatteetaten.....	27
4.3.1 Omorganisering av SITS.....	27
4.3.2 Fornyelse av IT-funksjonen (FORIT).....	27
4.3.3 I-plattform.....	28
4.3.4 NUPAS.....	29
4.4 Fagsystemene i Dokumentprosjektet.....	29
4.5.2 Sofie.....	29
4.5.3 MVA.....	30
4.5.4 Print.....	30
5. Forskningstilnærming.....	31
5.1 Forskningsperspektiv og filosofiske betraktninger.....	31

5.2	Forskningsdesign	32
5.2.1	Valg av case	32
5.3	Forskningsstrategi og metodisk tilnærming	33
5.3.1	Datainnsamling	34
5.3.2	Dataanalyse	36
5.4	Forskningens kvalitet	37
5.4.1	Vurdering av forskningens kvalitet	38
5.4.2	Bruk av teori i studien	39
5.4.3	Min rolle som forsker	39
6.	Resultater og analyse	41
6.1	Dagens prosjekter og status	41
6.1.1	Dokumentprosjektet	42
6.1.2	FDOK	45
6.1.3	MDOK	47
6.1.4	SDOK	49
6.1.5	SporDok	49
6.2	Forventede effekter av Dokumentprosjektet	50
6.2.1	Skatteetaten	50
6.2.2	Skatteetatens Service- og IT-partner (SITS)	51
6.2.3	Print-miljøet	51
6.2.4	Fagsystemene	51
6.2.5	FDOK	51
6.2.6	Skatteyttere	52
6.3	Utfordringer underveis i prosjektene	52
6.3.1	Sikkerhet	52
6.3.2	Ytelse	52
6.3.3	Kompetanse	52
6.3.4	Koordinering mellom prosjektene	54
6.3.5	Kommunikasjon	55
6.3.6	Parallelle prosjekter	57
6.3.7	Bruk av konsulenter i viktige roller	58
6.3.8	Prosjekt- og systemutviklingsmetode	59
6.3.9	Kompleksitet	60
6.3.10	Oppsummering av sentrale utfordringer	61

7. Diskusjon	63
8. Konklusjon og implikasjoner.....	69
8.1 Konklusjon	69
8.2 Implikasjoner for praksis	69
8.3 Implikasjoner for forskning	70
8.4 Begrensninger ved studien og forslag til videre forskning	70
9. Referanser	71

Tabelloversikt

Tabell 1 Karakteristikk ved arkitektturnivåene (Ross, 2003).....	8
Tabell 2 SOA modenhetsmodell i tabellform (Hirschheim et al., 2010).....	10
Tabell 3 Ontologi (Walsham, 1995).....	31
Tabell 4 Epistemologi (Klein & Myers, 1999; Walsham, 1995)	31
Tabell 5 Deltakere i studien.....	35
Tabell 6 Forskningens kvalitet etter Klein & Myers (1999) prinsipper.....	38
Tabell 7 Sentrale funn i studien.....	61

Figuroversikt

Figur 1 Oppgavens oppbygging	2
Figur 2 OASIS tjenestekonsepser (OASIS, 2006)	4
Figur 3 SOA Domain model (BEA, 2005).....	6
Figur 4 SOA modenhetsmodell (Hirschheim et al., 2010)	9
Figur 5 View-basert modell. (Tran et al., 2007).....	11
Figur 6 BEAs domenemodell (BEA, 2005).....	13
Figur 7 BEA Domain Model for SOA (BEA, 2005).....	14
Figur 8 BEAs figur over kostnader ved SOA-program (BEA, 2005)	15
Figur 9 Oracles SOA Maturity Model Domains (Oracle, 2010).....	16
Figur 10 Styring av SOA (Oracle, 2007).....	18
Figur 11 Six Steps to Successful Governance (Oracle, 2007).....	18
Figur 12 Balansert styring av SOA (Erl, 2011)	19
Figur 13 Scrum Methodology. (Schwaber, 1987)	23
Figur 14 Skatteetatens organisering (Skatteetaten, 2012)	25
Figur 15 Organisering Skattedirektoratet (Skatteetaten, 2012)	26
Figur 16 Components of Data Analysis (Miles & Huberman, 1994).....	36
Figur 17. Den hermeneutiske sirkel (Creswell, 1998).....	37
Figur 18 Dokumentprosjektets organisering i første fase	41
Figur 19 Dokumentprosjektets organisering i siste fase	42
Figur 20. Forenklet bilde av Skatteetatens portefølje (Staverløkk, 2009)	60

1. Innledning

Tjenesteorientert arkitektur (SOA) er en IT-arkitektur som er realisert gjennom løst knyttede tjenester med klare grensesnitt og skjult intern logikk (Hirschheim et al., 2010). Dette er en forholdsvis ny gren innen IT-infrastruktur. Tanken bak SOA er at man skal utvikle tjenester som direkte støtter organisasjonens forretningsprosesser. Når forretningsprosessene endres skal man relativt enkelt kunne endre koblingene mellom tjenestene. På denne måten kan man oppnå økt smidighet for organisasjonen (BEA, 2005). En smidig organisasjon vil lettere kunne tilpasse seg endrede konkurranseforhold i et globalt marked (Weill, 2002). For å oppnå en slik smidighet er man avhengig av å ha en tett sammenkobling mellom forretningsstrategien og IT-strategien gjennom SOA.

Skatteetaten er midt i et løp hvor IT-funksjonen fornyes gjennom å innføre en tjenesteorientert arkitektur (Hustad og Staverløkk, 2013; Staverløkk, 2009). Skatteetaten har fra tidligere en kompleks IT-arkitektur bestående av et bredt spekter av fagsystemer som har vært utviklet og vedlikeholdt som silo-applikasjoner. Mange av disse fagsystemene kommuniserer via flere ulike former for systemintegrasjoner.

Skatteetaten skal nå implementere en ny dokumentløsning i flere av sine fagsystemer. Dette skal realiseres i form av en felleskomponent som utgjør et felles grensesnitt mot en sentral printløsning og brukertjenester ute i fagsystemene.

Det er forsket lite på SOA-governance og hvordan dette forholder seg til innføring av spesifikke fellestjenester (Joachim, 2011). I denne studien følges Dokumentprosjektet i et sosio-teknisk perspektiv for å studere hvilke faktorer som spiller inn på resultatet under innføring av en fellestjeneste i Skatteetatens komplekse IT-arkitektur.

1.1 Motivasjon

I denne oppgaven har jeg fulgt fire SOA-prosjekter i Skatteetaten for å forstå hvilke faktorer som spiller inn når man skal utvikle en fellestjeneste i en tjenesteorientert arkitektur. Disse har fellesbetegnelsen Dokumentprosjektet og omfatter fire delprosjekter. Jeg har fulgt dette prosjektet i praksis for å kunne generere kunnskap om hvilke utfordringer man kan møte når man innfører en fellestjeneste som griper inn i store deler av en organisasjon. Studien er basert på et sosio-teknisk perspektiv for å kunne forstå hvordan samspillet mellom informasjonsteknologi og menneskelige faktorer innen en organisasjon kan påvirke prosjektprosessen og prosjektets utfall.

Tjenesteorientert arkitektur er et forholdsvis nytt og meget spennende fagfelt innen informasjonssystemer. SOA har blitt tatt i bruk av flere bedrifter (Baskerville et al., 2010), men det har foreløpig vært gjort lite empirisk forskning på fagområdet.

Jeg ønsker å forstå mer om hvordan man styrer og utvikler en fellestjeneste i en kompleks systemutviklingskontekst, og hva som må til for å lykkes. Slike prosjekter vil nødvendigvis gripe inn i og påvirkes av mange fagfelt i organisasjonen. Jeg ønsker derfor å øke kunnskapen på dette feltet og studere hva som påvirker systemutvikling av tjenesten. Jeg håper så å kunne benytte denne kunnskapen til å komme med anbefalinger om hvordan fremtidige prosjekter kan legges opp for å oppnå et optimalt resultat.

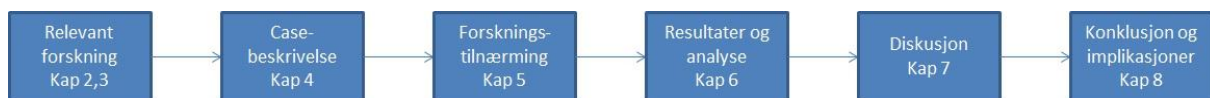
1.2 Problemstilling og forskningsspørsmål

Mine forskningsspørsmål har i denne studien vært:

1. Hvordan gjennomføres SOA-prosjekter i praksis?
2. Hvilke utfordringer er sentrale ved innføring av fellestjenester i en tjenesteorientert arkitektur?
3. Hvordan kan en organisasjon ta lærdom av denne type prosjekter?

1.3 Leseveiledning

Studien benytter en tradisjonell oppbygging for å lede leseren gjennom studien (se figur 1).



Figur 1 Oppgavens oppbygging

I kapittel 2 og 3 presenteres sentral litteratur fra aktuelle fagfelt som har vært viktige for studien. Først introduseres tjenesteorientert arkitektur og forskning på IT-arkitektur og modenhetsnivåer for å se hvordan dette vil påvirke en organisasjons evne til å innføre fellestjenester. Deretter presenteres forskning om systemutvikling i en kompleks kontekst og hvordan dette kan ses i et sosio-teknisk perspektiv. Sentrale faktorer man må fokusere på ved styring av systemutviklingsprosjekter i et tjenesteorientert perspektiv presenteres via rammeverk for innføring av tjenesteorientert arkitektur.

I Skatteetaten har man benyttet smidige systemutviklingsmetoder. Dette vil påvirke hvordan man leder og styrer et systemutviklingsprosjekt. Forskningslitteratur innen smidig systemutvikling presenteres for å forstå tankene bak, og hva sentrale forskere og premissleverandører mener man skal fokusere på for å lykkes.

I kapittel 4 introduseres studiens caseorganisasjon. Først beskrives Skatteetaten overordnet, før det fokuseres mer på etatens SOA-program og sentral prosjekthistorikk. For å presentere konteksten for Dokumentprosjektet er det viktig å kjenne til en del andre prosjekter som har vært gjennomført i Skatteetaten. Til sist i kapittel 4 beskrives de enkelte fagsystemene som har vært involvert ved innføring av en felles dokumentløsning.

I kapittel 5 forklares studiens forskningstilnærming gjennom å beskrive forskningsperspektiv, forskningsdesign, forskningsstrategi, samt en vurdering av forskningens kvalitet foretas.

I kapittel 6 presenteres resultatene fra studien og en analyse av disse. Dokumentprosjektet består av fire delprosjekter. Resultatene presenteres ved først å forklare gangen i de ulike prosjektene og hvilke sentrale hendelser som har påvirket resultatet. Så beskrives de forventede effekter dokumentløsningen vil gi. Sist i kapittelet presenteres de mest sentrale utfordringene man har støtt på i prosjektet og hvordan disse forholder seg til sentral forskning.

I kapittel 7 diskuteres hovedfunnene fra studien og hvordan disse forholder seg til tidligere forskning.

I kapittel 8 presenteres konklusjoner fra studien og implikasjoner for praksis og forskning.

2. Relevant tidligere forskning

I dette kapitlet presenteres relevant tidligere forskning innen tjenesteorientert arkitektur og SOA governance.

Først introduseres begrepet SOA og hvordan dette forholder seg til tjenester og prosesser. Deretter presenteres IT-arkitektur og modenhetsmodeller relatert til organisasjoners IT-arkitektur. I delkapittel 2.3 presenteres forskning på kompleksitet i organisasjonens IT-arkitektur og det sosio-tekniske perspektiv, og hvorfor dette er passende for å studere og forstå organisasjoners IT-implementasjon. Avslutningsvis presenteres sentrale rammeverk for implementasjon av SOA og governance av et SOA-program. Disse forklarer viktige momenter som man bør fokusere på i et SOA-program og hvordan dette vil påvirke organisasjonen.

2.1 Hva er SOA?

I 2009 var det flere som opplevde at SOA var inne i en liten krise. Det var uenighet i IT-miljøet om SOA virkelig hadde noe for seg, eller om det bare var en ny «hype» som ville forsvinne like raskt som den kom. Thomas Erl og andre sentrale aktører innen SOA var enige om at det var uklarerheter i forhold til hva SOA egentlig var. De kom frem til at man burde utarbeide et manifest lignende The Agile Alliances «Agile Manifesto» (Fowler et al., 2001). The Agile Manifesto fungerer som en ledestjerne i forhold til hvilke prinsipper som ligger til grunn for agil programmering.

Tanken utviklet seg litt over tid og i forbindelse med 2nd International SOA Symposium i Amsterdam oktober 2009 fikk man samlet en gruppe sentrale premissleverandører for SOA som skulle opprette «The SOA Manifesto» (Arsanjani et al., 2009; Erl, 2011). Selv om disse verdiene og prinsippene ikke er vitenskapelig fundamentert, så gir de likevel et godt innblikk i hva sentrale skikkelser mener SOA handler om. I løpet av symposiet kom gruppen frem til 6 verdier og 14 prinsipper som de mener beskriver kjernen i hva SOA skal være.

Verdier (fritt oversatt fra SOA Manifesto):

1. Forretningsverdi foran teknisk strategi
2. Strategiske mål foran prosjekt-spesifikke fordeler
3. Iboende interoperabilitet foran spesialtilpasset integrasjon
4. Delte tjenester foran spesial-designede implementasjoner
5. Fleksibilitet foran optimalisering
6. Evolusjonær tilpasning foran jakt på initiell perfektjon

Selv om gruppen verdsetter verdiene på høyre side, så foretrekkes verdiene til venstre.

Prinsipper (fritt oversatt):

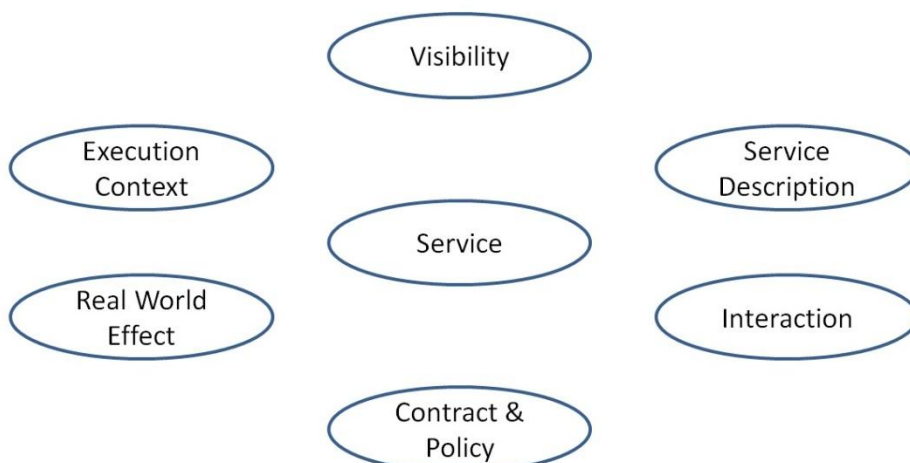
1. Respekter den sosiale strukturen og maktstrukturen i organisasjonen
2. Vær oppmerksom på at SOA til slutt krever endringer på mange nivåer
3. Graden av SOA adopsjon kan variere. Gjør tiltakene administrerbare og hold dem innenfor logiske grenser
4. Produkter og standarder vil alene hverken gi deg SOA eller det tjenesteorienterte paradigmet
5. SOA kan realiseres via en stor variasjon av teknologier og standarder

6. Etabler et felles sett av organisasjonsstandarder og policyer basert på bransje-, de facto- og samfunnsstandarder
7. Gå etter likhet på utsiden mens du tillater mangfold på innsiden
8. Identifiser tjenester gjennom samarbeid med interessenter fra både forretningssiden og teknologisiden
9. Maksimer bruken av tjenester ved å ta hensyn til både dagens og fremtidens grad av utnyttelse
10. Verifiser at tjenestene tilfredsstillor forretningens krav og mål
11. Utvikle tjenestene og organisasjonen på bakgrunn av bruk
12. Separer de forskjellige aspektene av et system som utvikler seg i ulik takt
13. Reduser implisitte avhengigheter og publiser alle eksterne avhengigheter for å øke robustheten og redusere følgene av endring
14. På hvert abstraksjonsnivå, organiser hver tjeneste rundt en sammenhengende og administrerbar enhet av funksjonalitet

“The goal of a Service-Oriented Architecture (SOA) is to enable organizations to realize business and technology advantages through a combination of process innovation, effective governance, and a technology strategy that revolves around the definition and re-use of services.” (BEA, 2005)

2.1.1 Hva er en tjeneste (service)?

Begrepet tjeneste (service) er et begrep som tolkes på mange forskjellige måter både i industrien og innen forskningen. Organisasjonen OASIS Open utga en referansemodell for tjenesteorientert arkitektur (2006) der de definerer en tjeneste i SOA som «mekanismene der behov og kapabiliteter bringes sammen». De uttaler videre at tjenesten er «en mekanisme som tilrettelegger for aksess til en eller flere kapabiliteter». En tjeneste er altså aksess til en kapabilitet til å utføre en operasjon som det er et forretningsmessig behov for å få utført. Figur 2 viser OASIS’ grunnmodell med tjenester og viktige egenskaper ved en tjeneste.



Figur 2 OASIS tjenestekonseppter (OASIS, 2006)

Hirschheim et al. (2010) uttaler i sin studie at «konsensus i markedet er at en tjeneste er innkapsling av spesifikk transaksjonell forretningsfunksjonalitet med en veldefinert grensesnittskontrakt som er

spesifisert ved hjelp av en åpen standard (for eksempel WSDL)». De skriver videre at en tjeneste er en softwarekomponent som kan aksesseres via internett og gir tilgang til klart spesifiserte funksjoner. En tjeneste gir tre klare fordeler; de kan gjenbrukes, de kan endres, forbedres og flyttes uten at det påvirker grunnfunksjonaliteten, og de tillater at man kan koble sammen applikasjoner og datakilder sømløst. Hirschheim et al. (ibid) undersøkte i sin studie hva organisasjoner med et SOA-program la i tjenester og andre begreper relatert til SOA. Det kom da frem at deres intervjuobjekter hadde en annen definisjon av tjenester enn det forskerne anså som konsensus i markedet. Intervjuobjektene definerte tjenester som alle softwarekomponenter som var pakket inn i et grensesnitt definert av industrien (for eksempel WSDL, DCOM, ASP.Net).

I denne studien er det lagt til grunn en definisjon som ligger opp mot Hirschheim et als definisjon der en tjeneste er en forretningsfunksjonalitet som er pakket inn og tilbys gjennom klart definerte grensesnitt.

2.1.2 Hva er en prosess?

Davenport et al. (1990) definerte prosess som *«et sett med logisk sammenhengende oppgaver utført for å oppnå et definert forretningsmål»*. De sier videre at en prosess har to viktige kjennetegn; at de har mottakere som tar i mot resultatet av prosessen og at de kan krysse organisatoriske enheter.

Ultimus (2003) definerer en prosess litt mer detaljert ved å si at en prosess er *«en sekvens av strukturerte eller semistrukturerte handlinger utført i serie eller parallelt av to eller flere individer for å oppnå et felles mål»*.

Michael Porter (1985) oppsummerer i sin «Value Chain Model» verdikjeden i en organisasjon og hvordan denne består av kjerneprosesser og støtteprosesser. Kjerneprosesser er selve verdiflyten der organisasjonen utfører sin kjernevirksomhet og sin verdiskaping. Støtteprosessene er alle aktiviteter som skjer rundt for å støtte opp om og legge til rette for verdiskaping. Det er alle disse forretningsprosessene, og spesielt kjerneprosessene, man har behov for at IT-infrastrukturen vår støtter på en best mulig måte i SOA.

Det finnes mange rammeverk som fokuserer på å få forretningsviden og IT-siden til å trekke i samme retning og støtte hverandre for å oppnå verdier. Balanced Scorecard (Kaplan & Norton, 1992) og COBIT (ISACA, 2012) er to eksempler på dette. I SOA søker man å oppnå dette ved en innretting mellom tjenestene og forretningsprosessene (BEA, 2005; Beimborn et al., 2009; Hirschheim et al., 2010). Man søker da ofte å definere de sentrale forretningsprosessene som gir organisasjonen mest verdi. Man starter med å lage tjenester som støtter disse og jobber seg ut mot de mindre sentrale tjenestene, som en såkalt middle-out tilnærming (Hirschheim et al., 2010). Målet er å få en IT-infrastruktur som støtter opp om forretningsstrategien slik at man oppnår smidighet og bedre resultater.

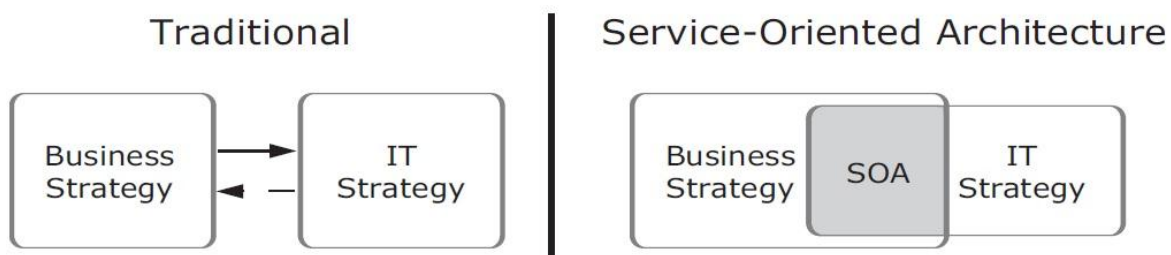
“If you automate a bad business process, you get a faster, more expensive, bad business process”
(Markus, 2004)

2.1.3 Tjenesteorientert arkitektur

Tjenesteorientert arkitektur, eller Service-Oriented Architecture (SOA), er et begrep og en tankegang som synes å bli tatt stadig mer i bruk. Hirschheim et al. (2010) definerer SOA ved å si at SOA er en IT-arkitektur hvor data og den logiske funksjonaliteten pakkes inn slik at bare input og output er synlig og kan brukes av andre. Med dette menes det at man deler opp IT-arkitekturen sin i mange tjenester i stedet for store applikasjoner, der tjenestene tilbyr funksjonalitet som andre tjenester og programvare kan benytte seg av ved å sende de korrekte inndata.

BEA (2008) forklarer at målet med SOA er å tilpasse organisasjonens IT-kapasiteter til de forretningsmessige målene og strategiene man har.

I et annet rammeverk utdyper BEA (2005) muligheten man har for å benytte tjenesteorientert arkitektur som et verktøy for å kunne fylle gapet som tidligere har eksistert mellom organisasjonens forretningsstrategi og organisasjonens IT-strategi (Figur 3). Ved å fokusere på forretningsstrategi og de forretningsprosessene som støtter opp om denne strategien kan man så skille ut de IT-tjenestene som muliggjør disse forretningsprosessene. Ved å skreddersy tjenestene rundt organisasjonens strategisk viktige prosesser vil man få en mer effektiv IT-infrastruktur. Hirschheim et al. (2010) sier at mye av interessen for SOA stammer fra at man ønsker å gå fra å utvikle store applikasjonspakker til å bygge tjenester som kan linkes direkte til forretningsbehovet. Baskerville et al. (2010) viser til at denne linken kan være vanskelig å oppnå, da det fortsatt ofte er uvant å inkludere forretningsiden i prosessen med å utvikle tjenester. Dette er en utfordring man må være bevisst på hvis man vil få full effekt av tjenesteorientering.



Figur 3 SOA Domain model (BEA, 2005)

Ordanini & Pasini (2008) definerer tre ulike perspektiver som man kan betrakte SOA fra. Disse er:

- Forretningsperspektiv, der tjenestene gir bedriften en kapasitet til å kunne gjøre forretninger med kunder og leverandører
- Teknologisk perspektiv, der SOA er en ny prosjektilosofi basert på modularitet, gjenbruk av tjenester og nye systemutviklingsmetoder
- IT-ledelsesperspektiv, der SOA utgjør en ny måte å designe applikasjonsporteføljen sin på

I SOA er det viktig å ta inn over seg disse tre ulike perspektivene når man styrer SOA-programmet sitt. Man ønsker å oppnå en sammenkobling mellom forretningsstrategien og IT-strategien, og da er det viktig at man har et helhetlig syn på hvordan man kan oppnå dette.

I følge Hirschheim et al. (2010) er det tradisjonelt sett 5 hovedårsaker til at man starter et SOA-prosjekt. Disse årsakene er at SOA legger til rette for økt forretningsmidighet, SOA dekker gapet mellom forretningsprosesser og IT-løsninger, SOA tilrettelegger for større grad av gjenbruk, SOA

tilrettelegger for inkrementell innføring av nye teknologier og SOA øker operasjonell fleksibilitet. I deres studie av organisasjoner som hadde implementert SOA var det derimot 5 andre hovedårsaker som ble nevnt. Disse var administrerbar infrastruktur, standardisering av ressurser, formelle metoder for gjenbruk, interaksjon med partnere og overordnet administrasjon av data i bedriften. Dette kan muligens forklares med at bedriftene var på et tidlig eller umodent stadium av sin implementasjon.

Tjenesteorientert arkitektur er et meget omfattende begrep og begrepet synes også å benyttes og defineres på flere ulike måter både innen forskning og i industrien. Dette oppdaget blant annet Baskerville et al. (2010) da de skulle gjennomføre en studie av to banker som innførte tjenesteorientert arkitektur. Den ene banken definerte tjenester på en måte som ligger utenfor den måten som man normalt benytter innen SOA, men banken definerte fortsatt løsningen som tjenesteorientert arkitektur.

Organisasjoner i dagens marked opplever stor konkurranse i et globalt marked. For å møte denne konkurransen må organisasjonene være i stand til å kunne endre seg og kontinuerlig handle hurtig på bakgrunn av endringer i miljøet (Fallmyr og Bygstad, 2011). SOA løser dette ved å koble forretningsstrategi og IT-strategi opp mot hverandre (BEA, 2005). Ved å gjøre dette vil både forretningsstrategi og IT-strategi være smidige slik at organisasjonen raskt kan endre seg i et konkurranseorientert marked.

For å oppnå full effekt av en tjenesteorientert IT-arkitektur er det viktig at man lar dette gjennomsyre hele organisasjonen (BEA, 2005). Man må klare å få koblet forretningsstrategien og IT-strategien tett opp mot hverandre slik at man får optimalisert forretningsprosessene sine. Så må man skape en IT-strategi og utvikle løst koblede tjenester som støtter opp om disse forretningsprosessene og gir den funksjonaliteten som organisasjonen er avhengig av for å kunne lykkes.

“Service-Oriented Architecture is an IT architecture where data and logic functionality are “black boxed” or encapsulated with only their input and output exposed for others to use” (Hirschheim et al., 2010).

2.2 IT-arkitektur

Jeanne Ross (2003) har skrevet en mye sitert artikkel om hvordan man kan skape en strategisk IT-arkitektur. Hun anbefaler her å analysere sin nåværende IT-arkitektur og utvikle seg inkrementelt mot målet der IT-arkitekturen støtter forretningsstrategien.

Ross påpeker at man ofte bruker begrepene arkitektur og infrastruktur litt om hverandre, der arkitekturen ofte er planen for den kommende infrastrukturen. På organisasjonsnivå blir IT-arkitektur definert som «*organiseringslogikk for applikasjoner, data og infrastrukturteknologier, slik de er samlet i policyer og tekniske valg med ønske om å legge til rette for bedriftens forretningsstrategi*».

En kompetanse innen IT-arkitektur er bedriftens evne til å skape et selvforsterkende miljø der man utvikler IT-kapasiteter som er tett knyttet opp mot forretningsstrategien og forretningsprosessene. For å utvikle dette kan man:

1. Definere bedriftens strategiske mål
2. Definere hvilke IT-kapasiteter som trengs for å nå disse målene

3. Definere policyer og tekniske valg for å utvikle disse IT-kapasitetene

Man kan dele inn IT-arkitekturer i fire forskjellige nivåer (Tabell 1). Disse vil variere i grad av modenhet og fleksibilitet for å kunne støtte forretningsstrategier og endringer i disse. De fire nivåene er:

1. Applikasjonssilo arkitektur
 - i. Arkitekturen består av individuelle applikasjoner som har hver sin arkitektur og oppbygging
2. Standardisert teknologi arkitektur
 - i. Arkitekturen griper gjennom hele organisasjonen og man har standardiserte teknologivalg og gjerne også sentralisering av ressurser
3. Rasjonaliserte data arkitektur
 - i. IT-arkitekturen utvides til også å standardisere data og prosesser
4. Modulær arkitektur
 - i. Hele organisasjonen følger globale standarder med applikasjoner, data og teknologikomponenter som er løst koblet mot hverandre. Dette gir en smidighet ved at man kan benytte globale standarder og likheter, men med lokale forskjeller. Dette nivået er tett knyttet opp mot tjenesteorientert arkitektur.

Tabell 1 Karakteristikker ved arkitekturnivåene (Ross, 2003)

	Application Silo	Standardized Technology	Rationalized Data	Modular
IT Capability	IT applications serve isolated business needs	Firm-wide technology standards	IT focused on wiring core process	Modules enable business model extensions
Key Management Innovation	Technology-enabled change management	Standardization and exception management, refresh	Recognizing essence of the business	Practices facilitating reusability
Business Case for IT	ROI of applications	Reduced IT costs; interoperability	Improved business performance; integration	Speed to market; Strategic agility
Locus of Control	Local control	Senior management support of CIO	Senior management, IT, and process leadership	Senior mgmt, IT, process, and local leadership
Key Governance Issues	Estimate, measure, communicate value	Establish (local/ regional/ global) standard setting, exception & funding processes	Determine core processes and funding priorities	Define boundaries for business experiments

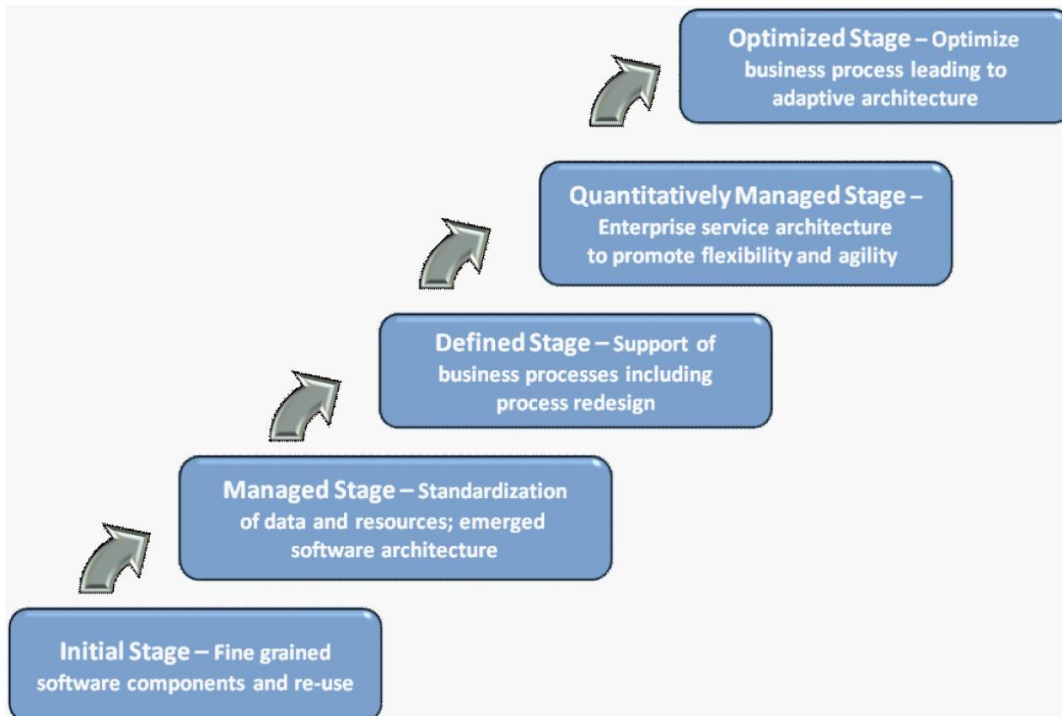
Hvert nivå har sine fordeler og sine måter å generere verdier for bedriften. Hvis man klarer å utnytte disse styrkene og utvikle organisasjonen fra nivå til nivå kan man gradvis skape en optimal og

fleksibel IT-arkitektur. For å oppnå dette har Ross kommet frem til seks erfaringer man bør ta med seg:

1. Fokuser på nøkkelforrettningsprosesser når man utvikler arkitekturen
2. Ikke hopp over eller jag gjennom modenhetsnivåer
3. Tenk på at komplekse organisasjoner kan ha flere arkitekturer og at disse kan være på forskjellige nivåer
4. Fang opp læring og erfaringer relatert til arkitekturen ved hjelp av styringsmekanismer
5. Fortsett dialogen, da arkitekturen aldri vil være ferdig
6. Sørg for å ha arkitekturkompetanse innad i organisasjonen

Ved å fokusere på IT-arkitekturen og utvikle seg inkrementelt i modenhetsnivåene vil man skape en tettere knytning mellom forretningsstrategien og IT-kapasitetene. Fallmyr og Bygstad (2011) gjennomførte en studie hvor de finner bevis for at et fokus på, og økt modenhet i, virksomhetsarkitekturen vil gi en mer smidig og fleksibel arkitektur som gjør organisasjonen bedre i stand til å gripe nye muligheter og skape nye verdier. Dette er helt i tråd med tjenesteorienteringens mål (BEA, 2005; Hirschheim et al., 2010; Oracle, 2010).

Hirschheim et al. (2010) har utviklet en SOA modenhetsmodell på bakgrunn av en studie de gjorde av 10 bedrifters erfaringer med SOA (Figur 4). Modellen deres er delt inn i 5 ulike modenhetsnivåer ved organisasjoners SOA-program. I deres studie fant de at mange av bedriftene de undersøkte fortsatt befant seg på tidlige modenhetsnivåer, og at effektene av SOA-programmet og knytningene mellom forretning og IT vil øke etter hvert som man beveger seg oppover på stigen.



Figur 4 SOA modenhetsmodell (Hirschheim et al., 2010)

Tabell 2 viser hvordan Hirschheim et al. bryter ned de ulike modenhetsnivåene innenfor bestemte dimensjoner. Dette illustrerer godt forskjellene mellom de ulike nivåene og hva som kjennetegner en organisasjon på hvert enkelt nivå.

Tabell 2 SOA modenhetsmodell i tabellform (Hirschheim et al., 2010)

Dimension	Maturity Level				
	<i>Initial</i>	<i>Managed</i>	<i>Defined</i>	<i>Quantitatively Managed</i>	<i>Optimized</i>
<i>View of SOA</i>	Fine-grained software components	Emerged software architecture	Support of business processes	Enterprise service architecture	Adaptive architecture
<i>Benefits & Metrics</i>	Promise of re-use; no metrics	Standardization of data and resources	Business process redesign	Agility and flexibility	Autonomic systems
<i>Business involvement</i>	Hidden from business	Services exposed as part of project costing	Services part of requirement capture	Organizational “service thinking”	Value-stream “service thinking”
<i>Methodology</i>	Minor adaptation of current S/W methods	Project-level service definition methods	Business service definition methods	Intra- and inter-organizational service definition	Value-chain service optimization
<i>Sourcing of services</i>	Largely derived from existing API’s	In-house development of new fine-grained software components	External sourcing possible by defined function	ESB to coordinate services	Full integration up and down the stack
<i>Governance</i>	Basic service definition policies at project level	Service policies managed by registry monitors	Full set of service policies and metrics in place	Business & IT governance metrics aligned	Policy feedback used to adjust delivery

IT-arkitekturen vil også endres og moderniseres etter hvert som organisasjonen modnes. Man implementerer tjenester på tvers av organisasjonen og vil få en mer modulær oppbygning av infrastrukturen sin. Dette legger igjen til rette for økt fleksibilitet og evne til å støtte forretningsstrategien.

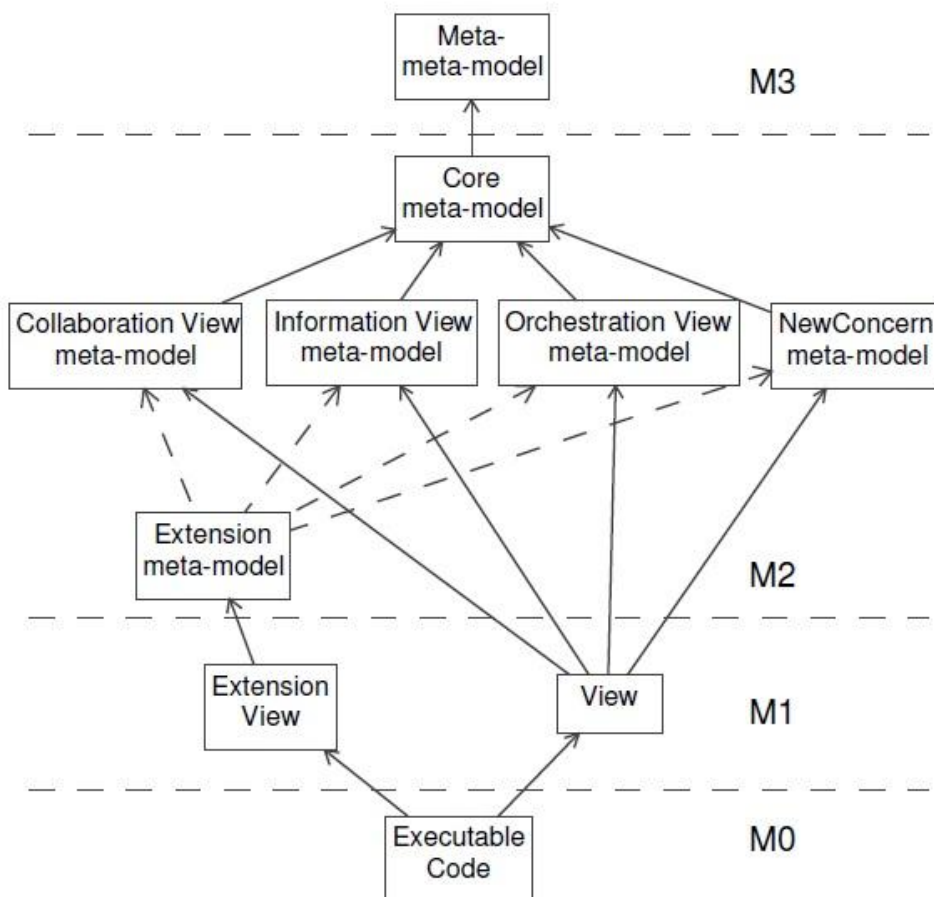
De to modenhetsmodellene til Ross (2003) og Hirschheim et al. (2010) fokuserer begge på modenhet ved virksomhetens IT-arkitektur. Selv om Ross sin modell ble utviklet før SOA ble introdusert så vil modenhetsutvikling av IT-arkitektur møte mange av de samme utfordringene. Både Ross og Hirschheim understreker at organisasjonen må ta seg tid til å modne IT-arkitekturen inkrementelt og se modenhet i hele organisasjonen under ett for å oppnå et best mulig resultat. Viktigheten av en inkrementell tilnærming og et helhetlig fokus støttes også av Lee et al. (2010) studie av kritiske suksessfaktorer ved implementasjon av SOA.

2.3 Kompleksitet og sosio-teknisk perspektiv

Det er flere forskere som har studert kompleksitet i IT-arkitekturen (Hirschheim et al., 2010; Ross, 2003; Tran et al., 2007). Samfunnet og våre organisasjoner har gjort oss veldig avhengige av avanserte informasjonssystemer. Organisasjoner som har eksistert i mange år vil ofte ha utviklet en fragmentert IT-infrastruktur bestående av systemer basert på mange ulike plattformer og systemutviklingspråk (Baskerville, 2010). Når man skal modne arkitekturen sin og introdusere tjenester i organisasjonen vil man ofte få komplekse situasjoner der man vil måtte forholde seg til et komplekst systemutviklingsmiljø (Ross, 2003). For å lykkes med systemutviklingsprosjekter i et

komplekst miljø kreves det muligens andre verktøy og andre fokus enn systemutvikling i mindre komplekse miljøer.

Tran et al. (2007) har forsket på systemutvikling i prosessdriven SOA. Deres utgangspunkt er at kompleksiteten i systemutviklingen er så stor at man trenger nye modeller for å organisere informasjonen og konteksten som påvirker systemutviklingen. De skriver at når antall elementer som er involvert i forretningsprosessarkitekturen (som prosesser, aktiviteter og tjenester) øker vil kompleksiteten i prosessutviklingen også øke ettersom sammenhenger, interaksjoner og dataoverføringer mellom elementene blir vanskeligere å styre. Utviklingsmiljøet blir så komplekst at man er nødt til å håndtere systemutviklingsprosjektene på en ny måte for å oppnå resultater. Deres forslag er et rammeverk hvor de introduserer en view-basert modell der man modellerer utviklingsmiljøet gjennom flere ulike representasjoner (Figur 5). Ved å angripe situasjonen fra flere ulike utgangspunkt vil man kunne fange opp de viktige elementene i den komplekse arkitekturen.



Figur 5 View-basert modell. (Tran et al., 2007)

Modellen baserer seg på en kjernemetamodell (core meta-model) som er kjernen i rammeverket. De andre modellene blir utvidelser av kjernemodellen. Kjernemodellen fanger opp viktige momenter som view, prosesser, tjenester og elementer. De andre modellene tar så for seg og modellerer hvert sitt aspekt av det helhetlige bildet. Samhandling (collaboration) handler om interaksjoner mellom de ulike prosessene. Informasjon (information) handler om informasjon som utveksles mellom prosessene. Orkestrering (orchestration) handler om aktiviteter og kontrollstrukturer mellom

prosessene. Ved å analysere prosessene og konteksten rundt gjennom de ulike view'ene er tanken at man vil få strukturert og fanget opp alle de viktige elementene som vil påvirke utviklingen av tjenester i den komplekse konteksten. Modellen kan også bygges ut ved å introdusere nye view, eller nye metamodeller, som fokuserer på andre aspekter i utviklingskonteksten.

Flere forskere argumenterer for at når man studerer informasjonssystemer kan man ikke frikoble innholdet i systemene fra konteksten de opererer i (Avgerou, 2001; Munkvold, 2000).

Implementering av informasjonssystemer i en kompleks kontekst kan best forstås ved å analysere de sosio-tekniske aspektene når teknologiske og menneskelige faktorer spiller sammen. Munkvold (2000) beskriver at det sosio-tekniske begrepet fokuserer på menneskene og felles optimalisering av det sosiale og det tekniske miljøet som viktige grunnelementer ved endring av organisasjoner og teknologi.

Avgerou (2001) identifiserer tre prinsipper man må følge for å kunne forstå det sosio-tekniske samspillet når man implementerer informasjonssystemer:

1. Teknologisk innovasjon må forstås i sammenheng med sosio-organisasjonsmessig endring
2. Analysen bør ikke bare ta inn over seg den lokale organisasjonen, men også den nasjonale og internasjonale konteksten
3. Analysen må inkludere både tekniske beslutninger og handlinger, og de kulturelle, sosiale og kognitive kreftene som skjer i endringsprosessen

Man kan ikke analysere endringer i informasjonssystemer uten også å fokusere på det komplekse samspillet som skjer mellom teknologi, og menneskene og organisasjonen som skal benytte teknologien. Det er derfor viktig å ha et bredere fokus når man forsker på informasjonssystemer. Man kan ikke bare tenke på teknologiske innovasjoner, men endringene i heterogene nettverk av institusjoner og mennesker der IT skal spille en viktig rolle.

2.4 Rammeverk for implementasjon av SOA

Implementering av SOA er et relativt umodent forskningsområde. Det har derfor vært nødvendig å fokusere på praktikerorientert litteratur som et supplement til forskningslitteraturen i denne studien. I denne delen vil jeg fokusere på noen sentrale rammeverk som er utviklet av ulike leverandører.

Det finnes veldig mange rammeverk tilgjengelig som forsøker å gi organisasjoner som skal innføre tjenesteorientert arkitektur en oppskrift over hvilke elementer man må ha fokus på for å lykkes. Her presenteres noen sentrale rammeverk i forhold til implementasjon av et SOA-program og SOA governance.

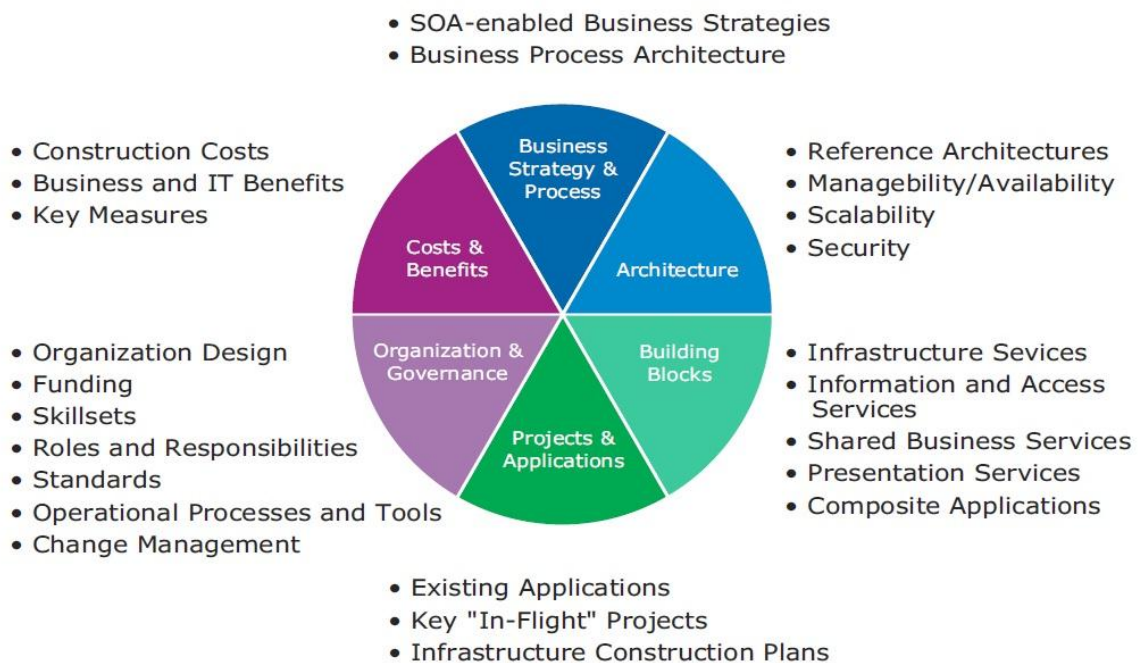
«The greatest benefit and return on investment with SOA occurs when companies stop trying to maximize investments locally (silos) and start maximizing all assets across the entire enterprise» (Oracle, 2007)

2.4.1 BEA domain model

BEA er en aktør som hadde lang erfaring med tjenesteorientert arkitektur og innføring av dette i bedrifter før de ble kjøpt opp av Oracle i 2008. BEA produserte flere praktikerorienterte rammeverk i forbindelse med sin satsing på SOA. BEA (2005) forklarer i et rammeverk bakgrunnen for sin domenemodell med at «ved å organisere organisasjonens IT rundt tjenester istedenfor applikasjoner kan man oppnå følgende effekter:

- Forbedret produktivitet, smidighet og hurtighet for både forretnings siden og IT
- Tillater IT å levere tjenester raskere og mer nærliggende forretnings siden
- Tillater forretnings siden å respondere raskere og levere optimale brukeropplevelser»

BEAs forslag til hvordan man skal oppnå dette er at man må fokusere på en balansert innsats mot 6 ulike domener for å oppnå en tjenesteorientert arkitektur (Figur 6).

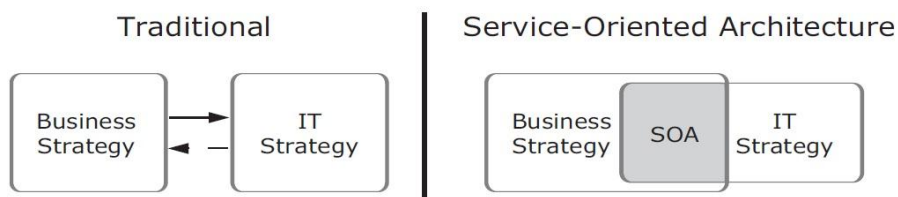


Figur 6 BEAs domenemodell (BEA, 2005)

De 6 domeneene er:

1. Business Strategy and Process

Tradisjonelt sett har ofte forretningsstrategi og IT-strategi vært sett på som to forskjellige ting. Forretnings siden analyserer organisasjonen og utvikler langsiktige strategier for å utvikle organisasjonen i ønsket retning. IT-strategien utvikles av IT-avdelingen og har gjerne et snevrere fokus på eksisterende systemer og benytter en kortere tidshorisont. Ved SOA er det viktig at IT-strategien knyttes opp mot organisasjonens forretningsprosesser og forretningsstrategi slik at man utvikler en smidig og samordnet strategi illustrert i figur 7. Hirschheim et al. (2010) fant at mange organisasjoner fortsatt ser SOA som et prosjekt for IT, men for å oppnå full effekt av SOA må man knytte forretning og IT sammen.



Figur 7 BEA Domain Model for SOA (BEA, 2005)

Thomas Erl (2011) nevner flere kritiske suksessfaktorer relatert til en SOA-implementasjon. En av disse er viktigheten av et tett samarbeid mellom forretningsanalytikere og teknologiarkitekter når man analyserer prosessene i organisasjonen og modellerer tjenester ut av disse. Spesielt viktig er deltakelsen fra forretningsanalytikerne og folk med kjennskap til den forretningsmessige siden av driften. Denne faktoren nevnes også av flere andre forskere og praktikere innen SOA, og må ses på som en meget viktig faktor (BEA, 2008). Dessverre kan det lett bli slik at SOA ses på som et IT-prosjekt, og man får aldri den riktige tilknytningen og forankringen av prosjektet.

2. Architecture

Arkitekturen definerer hvordan man leverer funksjonaliteten på en smidig måte i organisasjonen. Man må tenke på leveransene i form av tjenester som kan leveres én gang og benyttes av flere andre tjenester ved behov. En referansearkitektur gir muligheten til å definere de viktigste komponentene i SOA på et høyt nivå, og viser hvordan komponentene forholder seg til hverandre. En slik referansearkitektur kan brukes som et målkart der man flytter seg fra et silo-basert syn på applikasjoner til et felles tjenesteorientert syn (Ross, 2003).

3. Building Blocks

Alle de gjenbrukbare elementene man utvikler som en del av SOA-programmet vil være organisasjonens byggeblokker. BEA deler disse inn i software byggeblokker og organisatoriske byggeblokker. Software byggeblokker innbefatter kode, datamodeller, prosesser, tjenester, applikasjoner og komponenter. Organisatoriske byggeblokker innbefatter beste praksis, standarder og verktøy for å administrere og utvikle tjenestene. Byggeblokkene vil så til sammen utgjøre forretningsinfrastrukturen. Det er viktig å utvikle byggeblokkene inkrementelt. Man må også sørge for å gjenbruke byggeblokkene der hvor man kan det.

4. Projects and Applications

Basert på referansearkitektur og forretningsens mål bilde, så er det viktig å identifisere hvilke tjenester og prosesser som bør utvikles med tanke på hvilken effekt det tilbyr. BEA anbefaler å utvikle et «SOA Roadmap» der man spesifiserer hvilke tjenester som skal innføres og i hvilken rekkefølge for å gi en best mulig effekt i forhold til innsatsen som kreves. Man må få oversikt over eksisterende prosjekter og portefølje og så legge en plan for gjennomføringen. Det anbefales å gjøre dette som en inkrementell prosess i stedet for å prøve å tjenesteorientere hele organisasjonen over natten.

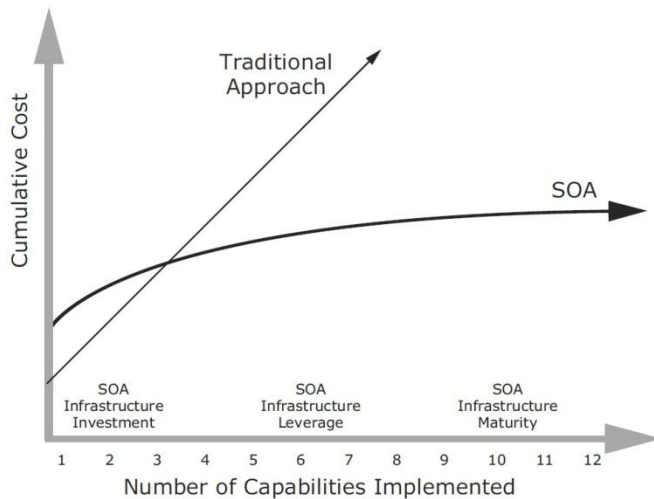
5. Organization and Governance

Å starte opp et SOA-program er en meget omfattende oppgave. Man skal reorganisere hele organisasjonen slik at man knytter forretningsstrategi og IT-strategi sammen og lager tjenester som støtter forretningsprosessene. For å lykkes i et så stort prosjekt er man

avhengig av at hele organisasjonen involveres i arbeidet. BEA uttaler at styring og utvikling av organisasjonen er selve fundamentet i domenemodellen deres. Man må fokusere på alle de følgende områdene: etterlevelse av standarder, målbilde for tjenestene, endringsledelse, fokus på gjenbruk, organisasjonsstruktur, kulturendring, utvikle ferdigheter, bruke beste praksis, og finansiering. For å lykkes må SOA-programmet forankres helt fra toppen og gjennomsyre hele organisasjonen.

6. Costs and Benefits

Å beregne kostnader og effekter av SOA kan være vanskeligere enn i et rent silo-basert miljø. Når man starter et SOA-program så er det viktig å være bevisst på at effektene begynner å melde seg etter år snarere enn etter måneder. Investeringene i starten vil bli høye, og i mange tilfeller vil det koste mer å utvikle fellestjenester enn å utvikle enkeltapplikasjoner. Forskjellen er at når man utvikler fellestjenester og begynner å gjenbruke disse så vil utviklingskostnadene avta i et SOA-program, slik figur 8 illustrerer. Videre er det essensielt å understreke at når IT-strategien blir mer smidig og i større grad er samordnet med forretningsstrategien vil hele organisasjonen og dens informasjonssystemer bli mer endringsdyktige.



Figur 8 BEAs figur over kostnader ved SOA-program (BEA, 2005)

Innenfor hvert av domeneene er det viktig å ha et felles fokus som leder mot en felles innsats for å bygge en tjenesteorientert infrastruktur. Det understrekes at domeneene i noen tilfeller vil overlappe hverandre, men at hovedhensikten er å oppnå en balansert tilnærming der hele organisasjonen trekker i samme retning. Målet er at man skal ha en IT-infrastruktur og en IT-strategi som støtter opp om forretningsstrategien og forretningsprosessene.

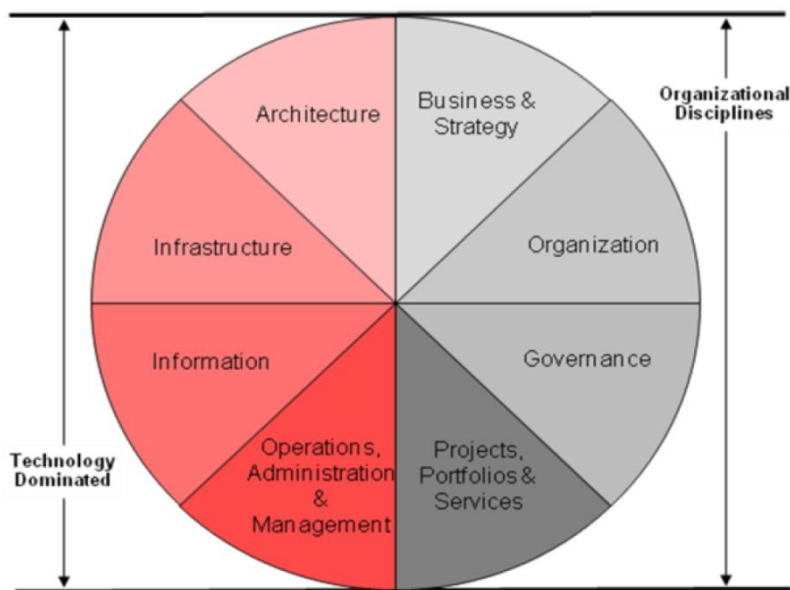
I en studie i Skatteetaten (Staverløkk, 2009) kom forskeren frem til at Skatteetaten hadde et SOA-program som hadde implementert mange av de kritiske suksessfaktorene identifisert innenfor hvert domene. Det var fortsatt en del områder hvor man kunne forbedre seg, eller der man var helt i startgroppen.

2.4.2 Oracle SOA Maturity Model

Oracle hadde lenge jobbet med tjenesteorientering og kjøpte opp BEA Systems i 2008 (Oracle, 2008). I september 2010 ga Oracle ut et rammeverk med tittelen «SOA Maturity Model – Guiding and

Accelerating SOA Success» (Oracle, 2010). I denne modenhetsmodellen kategoriseres organisasjonens kapabiliteter i 8 forskjellige domener illustrert i figur 9. Disse domenenene er:

1. Business & Strategy
2. Architecture
3. Infrastructure
4. Information
5. Projects, Portfolios & Services
6. Operations, Administration & Management
7. Organization
8. Governance



Figur 9 Oracles SOA Maturity Model Domains (Oracle, 2010)

Det finnes mange likheter mellom Oracles 8 domener og de 6 domenenene som BEA (2005) hadde i sin domenemodell, men Oracle utdyper en del av tankene fra BEAs modell. Oracle understreker i likhet med BEA viktigheten av å ha et balansert fokus på alle domenenene, og at domenenene i noen grad vil overlappe hverandre. Modenheten innen hvert domene vil totalt sett fortelle innen hvilke domener man bør legge mest trykk for å øke organisasjonens modenhet.

Selv om man ikke skulle ønske å foreta en full modenhetsstudie av organisasjonens SOA-program så inneholder fortsatt modellene mange interessante poenger for personell som er involvert i SOA-programmet. Som Oracle oppsummerer artikkelen sin med så vil SOA kunne redusere integrasjonskostnader, øke gjenbruk av ressurser, øke organisasjonens smidighet og gjøre det enklere å overholde regelverk. Oracle mener at man vil oppnå alle disse målene ved å følge deres beste praksis på en systematisk og helhetlig måte i organisasjonen.

“Service- Oriented Architecture (SOA) is not just another product or technology to be added to the IT environment. Rather, SOA requires significant and often fairly far-reaching changes to truly reap all the benefits of adopting SOA. To successfully adopt SOA, a company must create a plan that addresses the full extent of the changes required for SOA” (Oracle, 2010).

2.4.3 SOA governance

Governance, eller styring, er et begrep som går igjen når man snakker om å innføre et SOA-program. BEA (2005) går så langt som å si at hvis man undersøker mislykkede forsøk på gjenbruk av IT-ressurser så vil man i de fleste tilfeller kunne finne at mangel på styring av organisasjonen og programmet var en viktig faktor for utfallet. BEA uttaler at styring av SOA, og en organisasjon som støtter opp rundt dette, er selve fundamentet i deres domenemodell. Hirschheim et al. (2010) viser til at det ledende synet i markedet er at SOA og tjenestene må kontrolleres sentralt ved at man har en sentralisert styringsarkitektur. Lee et al. (2010) identifiserer kritiske suksessfaktorer relatert til SOA-implementasjoner i kategoriene governance, prosesser og teknologi.

Peter Weill (2004) sier at styring (governance) av IT-ressursene er essensielt fordi dette i stor grad påvirker effekten man får ut av investeringene man gjør i IT. Han har funnet at organisasjoner som yter best innen sine forretningsområder aktivt søker gevinster fra IT-ressursene ved å:

1. Synliggjøre forretningsstrategiene og hvilken rolle IT spiller for å nå dem
2. De måler og leder investeringene innen IT og hvilken effekt man får ut av dem
3. De styrer praksis i organisasjonen slik at IT tilpasses forretningsstrategien
4. De ansvarliggjør personell som skal lede endringene som gir resultater fra IT-ressursene
5. De lærer fra hvert prosjekt og blir bedre til å dele og gjenbruke IT-ressurser

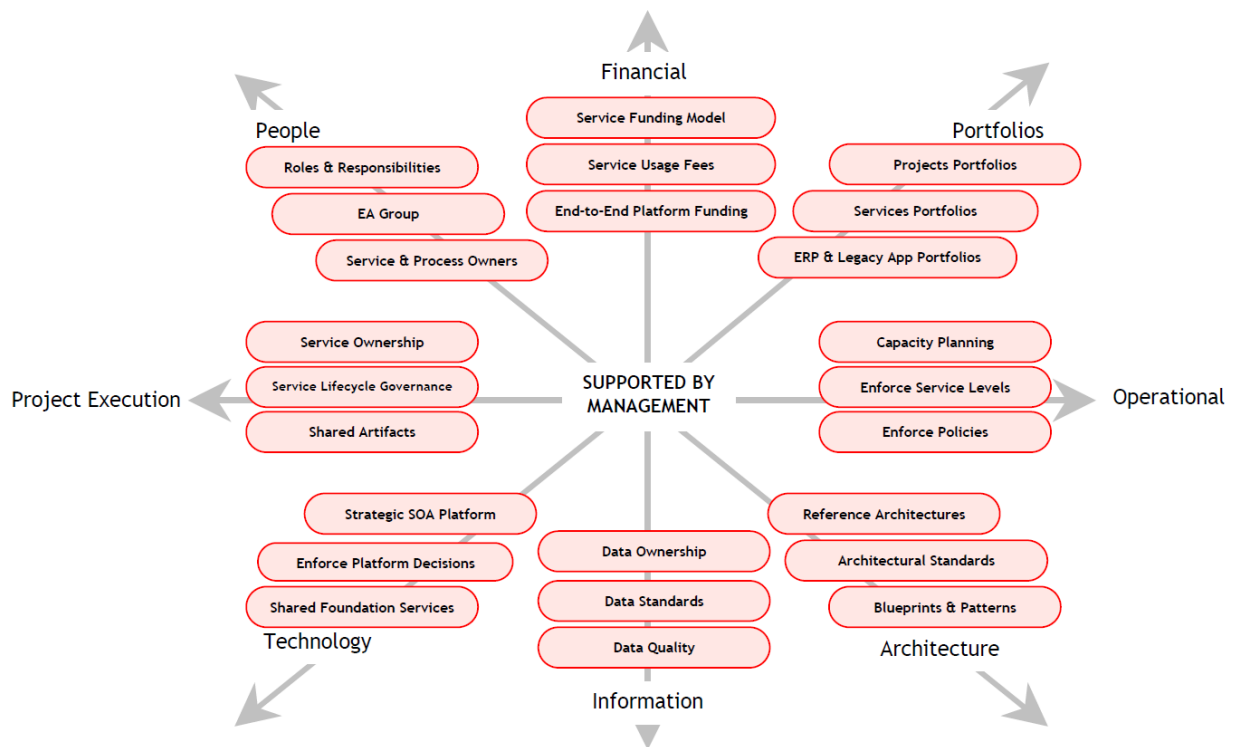
Flere av disse tankene ligger tett opp mot BEA (2005) sine råd om å knytte IT-strategier og forretningsstrategier opp mot hverandre.

Weill (2004) definerer IT-styring som å *«spesifisere rammeverket for å kunne ta beslutninger og ansvarliggjøring for å oppmuntre til en hensiktsmessig bruk av IT»*

Oracle (2007) viser til Paolo Malinverno i Gartner Group, som i 2010 sa at mangel på fungerende styringsmekanismer vil være den vanligste grunnen til at SOA-prosjekter feiler. Oracle (ibid) mener at styring av SOA handler om å levere på sine forretnings- og SOA-målsetninger. Man må koble investeringene knyttet til SOA mot forretningsmålene, prøve å begrense risikoen forbundet med SOA, og knytte dette opp mot organisasjonens overordnede styring av IT.

Oracle definerer styring av SOA som en effektiv sammenkobling mellom policyer, beslutningstakere og prosesser. Hvem skal ta hvilke avgjørelser på hvilken måte og hvordan skal de implementeres.

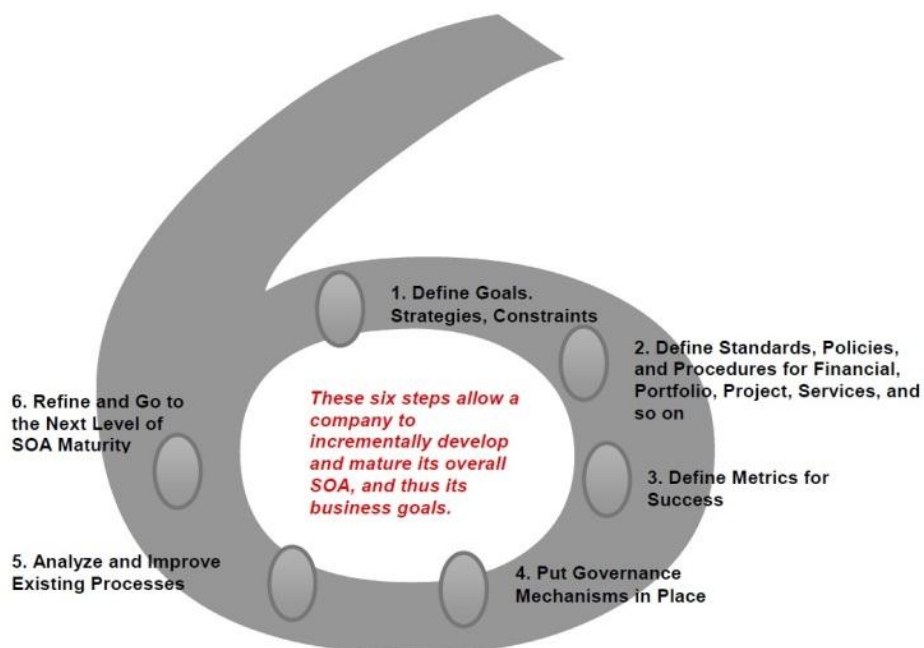
Når man så utvikler sine policyer og bestemmer hvordan man skal utvikle og benytte SOA i organisasjonen så er det viktig å ha et balansert fokus og gripe alle aspekter av organisasjonen og SOA-programmet. Oracle har laget en illustrerende figur som viser hvilke akser de ser på som essensielle i et styringsprogram (Figur 10).



Figur 10 Styring av SOA (Oracle, 2007)

Dette synet på styring av SOA, og viktigheten av å involvere hele organisasjonen, stemmer godt overens med det som synes å være konsensus. Både Marks et al. (2006) og Liegl (2007) fokuserer på at SOA er noe som griper om hele organisasjonen. Aksene i modellen stemmer godt overens med domeneene i Oracles egen domenemodell (2010), og BEAs domenemodell (2005).

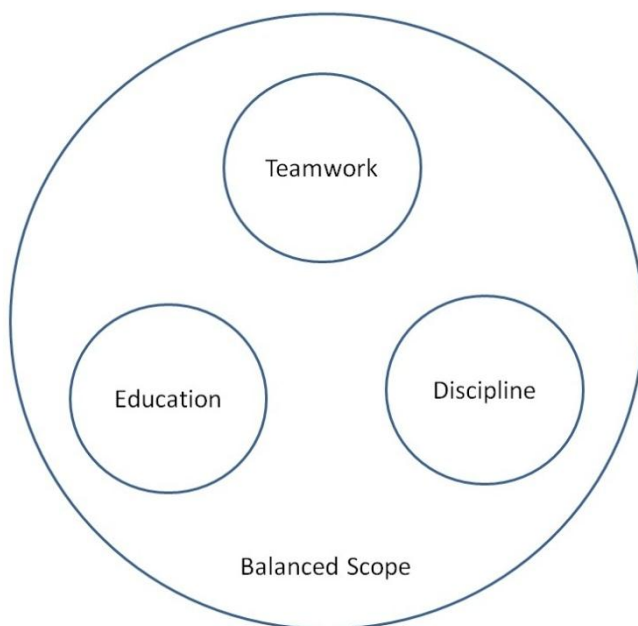
Oracle understreker at innføring av SOA bør gjøres som en inkrementell prosess for å oppnå et godt resultat. De anbefaler at organisasjonen styrer sitt SOA-program i seks steg, illustrert i figur 11.



Figur 11 Six Steps to Successful Governance (Oracle, 2007)

Oracle oppsummerer sin modell med å understreke at SOA vil være en del av organisasjonens virksomhetsarkitektur og bør knyttes tett opp mot bedriftens strategiske forretningsmål og forretningsprosesser. Det er viktig at man innretter og justerer innsatsen innen alle de åtte aksene, arkitektur, informasjon, teknologi, prosjektutførelse, mennesker, finansiering, porteføljer og operasjoner. Ved å styre SOA-programmet og ha forretningsmålene og understøttelse av disse i fokus kan man inkrementelt bygge en tjenestestruktur som er smidig og gir direkte støtte til organisasjonens sentrale forretningsprosesser.

Thomas Erl, en av initiativtakerne til «The SOA Manifesto» (Arsanjani et al., 2009), har gitt ut flere bøker om tjenester og tjenesteorientert arkitektur. I sin bok «SOA Governance» (2011) identifiserer han 4 pilarer eller kritiske suksessfaktorer man må ha fokus på i forbindelse med styring av SOA-program. Disse er teamarbeid, utdanning, disiplin og et balansert skop mellom disse (Figur 12).



Figur 12 Balansert styring av SOA (Erl, 2011)

Teamarbeid er spesielt viktig ved tjenesteorientering. Ved vanlige silo-baserte systemutviklingsprosjekter kreves det normalt sett kun at man samarbeider internt i prosjektgruppen. Ved et tjenesteorientert systemutviklingsprosjekt kreves det at man samarbeider på tvers av prosjektgruppene i langt større grad. Det stilles også høyere krav til samarbeidet, og manglende tillit mellom prosjektgrupper kan føre til at hele prosjektet feiler.

Utdanning er en annen nøkkelfaktor. For å kunne lykkes med å skape tjenester på tvers av organisasjonen er man avhengig av at man benytter samme retningslinjer, definisjoner, metoder og vokabular. Det kreves da en felles utdanning av personellet der man retter fokus mot en felles forståelse av hvilket mål man jobber mot.

Disiplin kreves for å utføre prosjektet på den måten man har lært at det skal gjøres. Man er avhengig av at alle i prosjektene er disiplinerte i måten de anvender kunnskapen sin for å få et vellykket resultat og en konsistent praksis i organisasjonen.

Et balansert skop skaper balanse mellom de samarbeidende prosjektteamene som har en felles forståelse og utdanning i hvordan tjenester skal utvikles, og gjør dette konsistent og disiplinert.

«Governance – whether it is financial, business, legal or IT – is about getting people to do the right things at the right time. In other words, it is about encouraging the behavior that will achieve your business goals” (Oracle, 2007)

2.4.4 Oppsummering rammeverk

De praktikerorienterte rammeverkene kan gi organisasjonen et godt utgangspunkt å jobbe etter når man skal innføre en tjenesteorientert arkitektur. Selv om rammeverkene ofte ikke er basert på forskning er de basert på erfaringer fra SOA-implementasjoner i mange ulike organisasjoner. De praktikerorienterte rammeverkene som er benyttet i studien (BEA, 2005; Oracle, 2007; Oracle, 2010) gir et godt bilde av helheten som må ligge til grunn for å lykkes med en satsing på en tjenesteorientert arkitektur. De viser at det er en omfattende og komplisert prosess som må involvere hele organisasjonen. Rammeverkene har et overordnet perspektiv og sier lite om konkrete prosjekter som innfører fellestjenester, men den helhetlige innsatsen vil gjøre seg gjeldende for hvert prosjekt i SOA-programmet. Tankene fra rammeverkene kan derfor gi gode indikasjoner på hva man bør fokusere på i konkrete prosjekter. Det vises videre at beste praksis fra de praktikerorienterte rammeverkene i mange tilfeller stemmer overens med hva flere forskere har kommet frem til (Liegl, 2007; Marks et al., 2006; Staverløkk, 2009; Weill, 2004).

3. Smidig systemutvikling

Deltakerne i Skatteetaten forteller at de benytter smidige metoder for systemutvikling. Mer spesifikt så forteller de at de benytter en tilnærming til Scrum i alle prosjektene. I hvilken grad de har implementert Scrum synes å variere noe, men alle sier de kjører Scrum-team og har daglige møter og produktkø. Det er derfor viktig å se litt på smidige metoder og Scrum spesielt for å undersøke hvilke utfordringer og fordeler dette kan skape i et tjenesteorientert perspektiv.

3.1 Smidige metoder

Alistair Cockburn er en anerkjent forfatter og en av initiativtakerne til «The Agile Manifesto» (Fowler et al., 2001). Han har skrevet flere bøker om smidig systemutvikling, blant annet sammen med Jim Highsmith. I «Agile Software Development – The Cooperative Game» (2007) beskriver de systemutvikling som et samarbeidsspill bestående av kreative løsninger og kommunikasjon utført med begrensede ressurser. Normalt sett vil man kanskje ikke se på systemutvikling som så veldig kreativt og med stor grad av kommunikasjon, men når man jobber i prosjektgrupper på større prosjekter og skal benytte smidig metode så er man helt avhengige av å være kreative og kommunisere godt.

Det finnes mange forskjellige smidige systemutviklingsmetoder, men generelt sett så regner man Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Crystal og Lean Software Development (LD) som de mest sentrale (Chow et al. (2008)).

Cockburn et al. (2007) skriver at «*Et velfungerende team, satt sammen av gode nok folk, klarer å gjennomføre et prosjekt nesten uansett hvilken prosess eller teknologi de benytter (selv om prosess og teknologi kan hjelpe eller hindre dem på veien)*».

Systemutviklerne er som oftest kreative og ønsker å utvikle noe selv. Derfor feiler ofte strategier om gjenbruk på grunn av effekten som forfatterne kaller «not-invented-here syndrome». Løsningen er å finne de riktige personene i prosjektet og gi dem oppdrag som passer til deres personlighet. Samtidig må man legge til rette for konsentrasjon og kommunikasjon i prosjektet.

Kommunikasjon kan man oppnå på mange forskjellige måter. Målet må være å få til så god informasjonsflyt som mulig i prosjektet. Dette kan økes ved å legge til rette for par-programmering og sørge for at systemutviklerne sitter så nært hverandre som mulig uten at det går ut over muligheten for å kunne konsentrere seg.

«Erg-sekunder» beskriver den tid og innsats som kreves for å kunne søke hjelp hvis man har et problem. Dette kan for eksempel reduseres hvis systemutviklerne sitter samlokalisert slik at man bare kan spørre ut i rommet om hjelp. Man har da det de kaller ekspert innen hørselsvidde. Man kan også oppsøke kontoret til den som har svaret, benytte telefon eller videokonferanse, men dette krever mer innsats og vil føre til svakere kommunikasjon.

Tillit er et annet viktig begrep ved effektiv kommunikasjon, og man samarbeider mye bedre når man kjenner og stoler på den andre parten. Det blir lettere å kommunisere og lettere å initiere

kommunikasjonen når man har opparbeidet tillit. Teambuilding kan være et veldig viktig grep for å skape denne tilliten og legge til rette for god kommunikasjon i teamene.

I forbindelse med kommunikasjon så vil geografisk lokalisering i systemutviklingsprosjektet være et viktig moment. Cockburn et al. (ibid) deler geografi inn i multisite, offshore og distribuert systemutvikling.

Multisite systemutvikling innebærer at man jobber på noen få lokasjoner der hver lokasjon har en komplett systemutviklingsgruppe som utvikler sub-systemer som er relativt frakoblet de andre. Nøkkelen her er å ha kompetente team på hver lokasjon, og at lederne fra hver lokasjon møtes ofte nok til å kunne dele visjon og forståelse av prosjektet.

Offshore systemutvikling innebærer outsourcing av systemutviklingen til programmerere på en annen lokasjon. Disse prosjektene feiler ofte på bakgrunn av at kommunikasjonen ikke blir god nok.

Distribuert systemutvikling innebærer at systemutviklerne er spredt på flere lokasjoner med få personer på hver lokasjon. Den beste måten å løse et slikt prosjekt på er å prøve å etterligne multisite systemutvikling ved å dele prosjektet opp i delprosjekter som hver lokasjon kan utvikle autonomt. Ellers så er tillit og grundig kommunikasjon virkemidler som kan gjøre situasjonen bedre.

For å tilpasse en smidig metode slik at den fungerer optimalt for sitt prosjekt så har Cockburn et al. (ibid) kommet frem til syv prinsipper man bør tenke på:

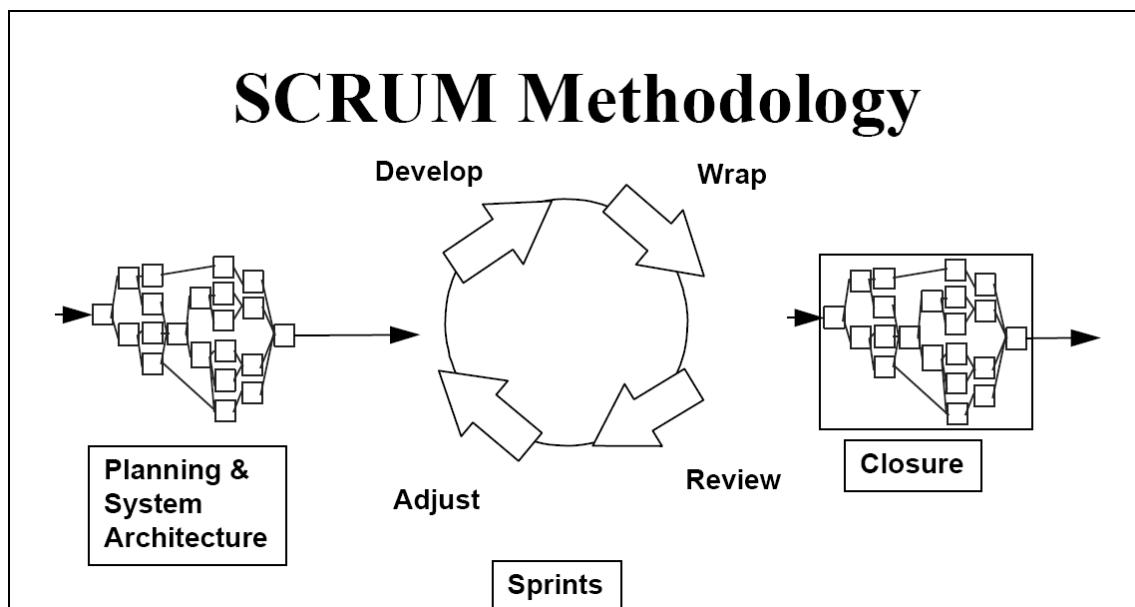
1. Interaktiv kommunikasjon ansikt til ansikt er den billigste og raskeste måten å overføre informasjon på
2. For tung metode er dyrt
3. Større team krever tyngre metoder
4. Flere kontrollmetoder passer best for kritiske prosjekter
5. Flere tilbakemeldinger og mer kommunikasjon reduserer behovet for leveranser underveis i prosjektet
6. Disiplin, ferdigheter og forståelse kan veie opp for prosesser, formalitet og dokumentasjon
7. Effektivitet kan ofres i aktiviteter som ikke er en del av flaskehalsen i prosjektet

Hvis man skal utvide et prosjekt er det bedre å tilføre mer kompetanse og kvalitet i gruppen ved å bytte ut uerfarne prosjektdeltakere med mer erfarne systemutviklere, enn ved å øke antall deltakere.

Chow et al. (2008) gjennomførte en studie der de testet kritiske suksessfaktorer hentet fra litteratur og forskning innen smidig systemutvikling. Suksessfaktorene ble testet ved kvantitativ metode. Man kom da frem til at i deres utvalg så var det kun seks suksessfaktorer som kunne verifiseres. Disse var en korrekt leveransestrategi, hensiktsmessige smidige software systemutviklingsteknikker, et dyktig utviklingsteam, god smidig prosjektledelse, hensiktsmessig utviklingsmiljø for teamet og involvering av kundene. Et systemutviklingsprosjekt kan derfor forvente suksess hvis man har et dyktig utviklingsteam som følger smidige systemutviklingsteknikker og benytter en korrekt smidig leveransestrategi.

3.2 Scrum

Ken Schwaber (1987) omtaler i sin artikkel "Scrum Development Process" Scrum som en systemutviklingsmetode. Han mener at den gamle utviklingstankegangen, som hevder at systemutvikling kan planlegges nøyaktig ved hjelp av plandrevne metoder, har spilt falitt. Bakgrunnen for hans Scrum modell er at systemutviklingsprosessen er en uforutsigbar, kreativ og komplisert prosess som mer ligner en inkrementell utvikling. Man må derfor ha en iterativ tilnærming til utfordringene i stedet for en metode som følger helt spesifikke steg. Denne ideen stemmer godt overens med Cockburns (2007) tanke om systemutvikling som et kreativt samarbeidsspill der kommunikasjonen er essensiell for å få alle til å trekke mot samme mål i en kompleks systemutviklingsprosess. Schwabers Scrum modell har siden den ble publisert fått mange tilhengere da den er en fleksibel og smidig systemutviklingsprosess der utviklingsteamet leder seg selv (Figur 13).



Figur 13 Scrum Methodology. (Schwaber, 1987)

Scrum deles inn i 3 hovedfaser. Systemutviklingen starter med en fase som kalles pregame, der planlegging og systemarkitektur/overordnet design er de to bestanddelene. Planleggingsfasen gir et utgangspunkt for den kommende systemutviklingen.

Fase nummer 2 er selve systemutviklingen og består av en rekke sprinter. Før hver sprint vil man ha en prioriteringsrunde der teamet bestemmer hva som skal utvikles innenfor denne relativt korte utviklingsperioden. Hver sprint består så av en iterativ prosess der man utvikler, pakker, evaluerer og justerer de utviklingsmålene man har bestemt seg for å nå i nettopp denne sprinten. Ved enden av sprinten skal alle i teamet ideelt sett ha utviklet de bestanddelene som teamet avtalte at de skulle utføre. Innenfor hver sprint har man jevnlig møter i utviklingsgruppen. Man møtes gjerne hver dag i et «standup-meeting» der teamet kort går igjennom hva de har utviklet den foregående perioden, hvilke utfordringer de eventuelt har og hva de planlegger å gjøre i den kommende perioden. Dette gir mulighet til å oppdatere teamet og omprioritere ressursene hvis det er behov for det. Sprintene går sekvensielt helt til systemet er ferdig utviklet, og dette blir da en iterativ prosess der man prioriterer målene og utvikler litt og litt av funksjonaliteten helt til man har et produkt som kunden blir fornøyd med.

I tredje fase, kalt postgame, gjennomføres en avslutning av hele systemutviklingen. Dette er en prosess der software frigjøres sammen med dokumentasjon.

Tanken er at selve systemutviklingsfasen tar inn over seg den kompleksiteten som ligger i et systemutviklingsprosjekt. I stedet for å tro at man kan planlegge et helt systemutviklingsprosjekt i detalj så åpner Scrum for flere iterative runder i utviklingen, der man løser små bestanddeler i hver Sprint og beholder en økt fleksibilitet i forhold til kontinuerlig å kunne identifisere og ta tak i eventuelle utfordringer.

I Scrum er tankegangen at utviklingsteamet i stor grad styrer seg selv. Systemutviklingsteamet består av systemutviklerne og Scrum master. Scrum master styrer teamet og planlegger møtene, men han har ikke avgjørelsesmyndighet. Tankegangen er at teamet skal lede seg selv i sprintene og utvikle systemet sammen. Kontakt med kundene er viktig fordi man slik får inn kundens behov (use cases), får mulighet til å prioritere hvilken funksjonalitet som er viktigst å få løst først, og får verifisert at denne funksjonaliteten tilfredsstiller kundens behov.

Cohn (2010) skriver at man i Scrum ønsker å involvere hele utviklingsteamet. Product backlog er de kravene og brukerhistoriene som beskriver hva man skal utvikle. Cohn oppfordrer til å søke å finne den riktige balansen i bruk av skriftlig dokumentasjon i prosjektene. Skriftlige kravspesifikasjoner kan ofte misforstås og mistolkes fordi de ikke er nøyaktige nok. Hvis man i stedet kunne diskutere kravene med dem som har kommet med behovet så ville man få en rikere kommunikasjon rundt behovet og avklare eventuelle uklarheter. Dette betyr ikke at man ikke vil at man skal dokumentere skriftlig, men at man er bevisst når og i hvilken form man gjør dette. Anbefalingen er å skrive brukerhistorier i stedet for krav. Brukerhistorier er korte historier som beskriver brukerens behov og konteksten dette behovet eksisterer i. På den måten blir behovet rikere beskrevet og gjør det enklere for systemutviklerne å forholde seg til og forstå.

Koordinering mellom Scrum team er et annet moment som Cohn påpeker at man må ha fokus på siden Scrum skaleres opp ved å bygge flere team heller enn større team. Det er én ting å lede ett enkelt Scrum team, men når flere team skal utvikle samtidig så stiller dette større krav til Scrum master eller prosjektleder. Løsningen er å holde «Scrum of Scrums»-møter der ett medlem fra hvert team møtes for å koordinere fremdrift. Møtene anbefales å holdes 2-3 ganger i uken, mot hver dag innad i teamet. Hvert team skal da presentere svar på 3 spørsmål:

1. Hva har mitt team gjort siden sist møte som kan påvirke andre team?
2. Hva vil vi gjøre innen neste møte som kan påvirke andre team?
3. Hvilke problemer har mitt team der vi kan trenge hjelp fra andre team?

Det anbefales også å synkronisere sprintene mellom teamene slik at man ikke overlapper hverandre. En slik overlapping vil gjøre det vanskeligere å koordinere arbeid på tvers av teamene siden alle vil være i forskjellige faser til en hver tid.

4. Casebeskrivelse av Skatteetaten

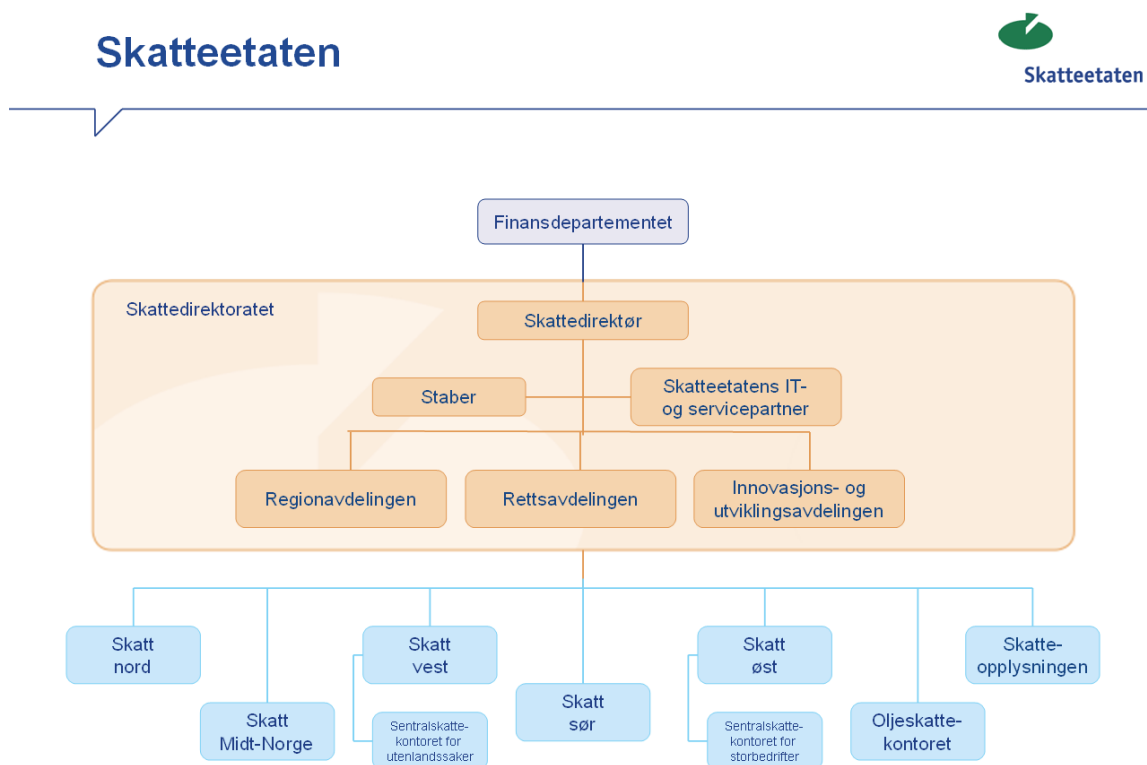
I dette kapittelet beskrives Skatteetaten som case. Kapittelet starter med en kort beskrivelse av Skatteetatens organisasjon og dens hovedoppgaver. Deretter sies det litt om SOA i Skatteetaten og hvordan dette er en viktig del av historikken og konteksten (Avgerou, 2001; Klein & Myers, 1999) til Dokumentprosjektet. Videre beskrives noen viktige prosjekter som har blitt gjennomført i forkant av Dokumentprosjektet. Disse prosjektene har vært en meget viktig del av konteksten og kan forklare noen av de utfordringene som har oppstått i Dokumentprosjektet. Til sist i kapittelet beskrives fagsystemene som er involvert i Dokumentprosjektet. Fagsystemene har hver sine fagområder og har forskjellige bruksområder. En beskrivelse av dette vil hjelpe leseren til å forstå behovene fagsystemene hadde.

4.1 Skatteetatens organisasjon

Skatteetaten er underlagt Finansdepartementet og er ansvarlig for at skatter og avgifter fastsettes og innbetales på riktig måte. Skatteetaten er ansvarlig for folkeregisteret og for at dette er oppdatert.

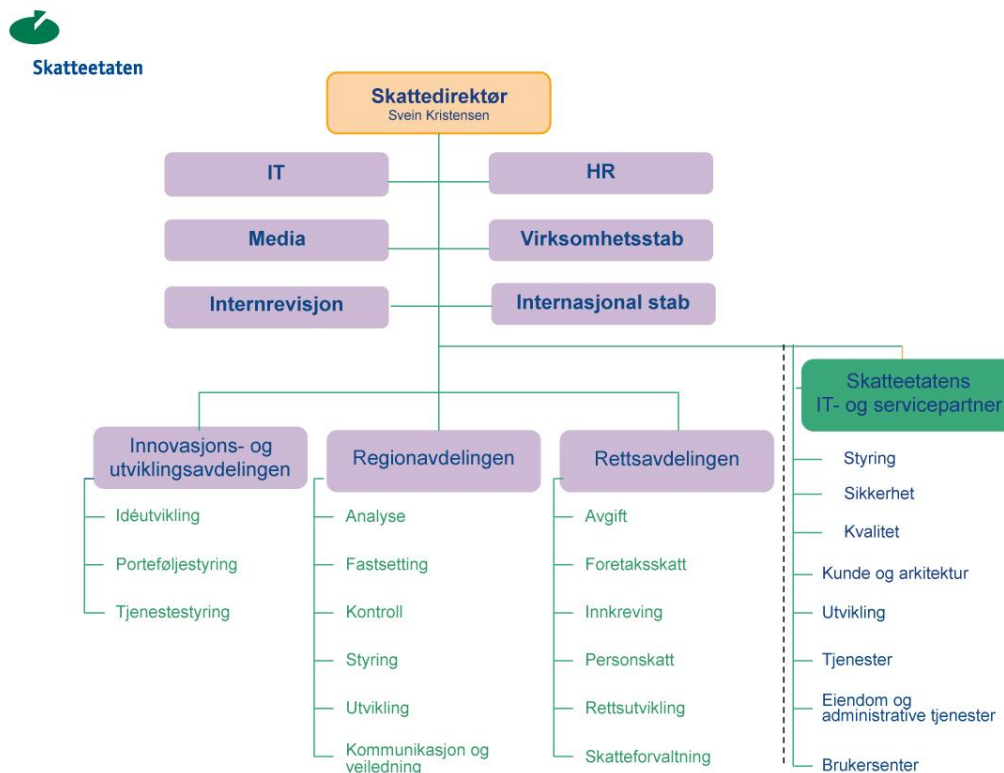
I 2011 behandlet man 4 115 500 selvangivelser og samlet inn 834,4 milliarder kroner i skatt. Det er i underkant av 6400 ansatte i Skatteetaten. Fra årsrapporten i 2011 kan man lese at utgiftene i forbindelse med større IT-prosjekter var på 300,9 millioner.

Skatteetaten er organisert med ett hovedkontor, Skattedirektoratet. I tillegg har man fem regioner: Skatt Nord, Skatt Midt-Norge, Skatt Øst, Skatt Vest og Skatt Sør. Hver region har flere skattekontorer. Figur 14 viser Skatteetatens organisering.



Figur 14 Skatteetatens organisering (Skatteetaten, 2012)

Skattedirektoratet består av det strategiske direktoratet med faglig og administrativ ledelse, og Skatteetatens IT- og Servicepartner. Figur 15 viser Skattedirektoratets organisering.



Figur 15 Organisering Skattedirektoratet (Skatteetaten, 2012)

Skatteetatens IT- og Servicepartner (SITS) har ansvar for utvikling og drift av Skatteetatens IT-portefølje. SITS er lokalisert i Oslo, Grimstad og Hamar. Direktøren i SITS inngår i Skattedirektørens ledergruppe. Denne studien ble gjennomført i SITS.

SITS er rammefinansiert med faste driftsmidler. Det kan komme prosjekter med ekstra ressurser hvis det er behov.

4.2 SOA i Skatteetaten

Prosjektet Fornyelse av IT-funksjonen ble startet i Skatteetaten i 2009 (se punkt 4.3.2). Prosjektet har til mandat å skulle modernisere Skatteetatens IT-infrastruktur.

Våren 2009 startet Virksomhetsarkitekturprosjektet som skal etablere en overordnet virksomhetsarkitektur og en tilhørende IT-reguleringsplan (Hustad og Staverløkk, 2013; Staverløkk, 2009). Prosjektet analyserte de eksisterende applikasjonene og plattformene og la grunnlaget for senere prosjekter i fornyelsen.

Det er nå utviklet en felles integrasjonsplattform (I-plattform) og man har utviklet en felles systemutviklingsplattform (Aurora). En sentral arkitekturenhet involveres i alle nye

systemutviklingsprosjekter og man har et eget porteføljestyre som prioriterer nye prosjekter og gir føringer og retningslinjer for gjennomføringen.

4.3 Tidligere prosjekter i Skatteetaten

Skatteetaten har gjennomført mange prosjekter for å utvikle sitt SOA-program. Noen av de viktigste har utviklet en egen integrasjonsplattform og en egen systemutviklingsplattform. Disse er utviklet som en del av Skatteetatens SOA-program og er viktige komponenter som har lagt føringer for Dokumentprosjektet. For å forstå konteksten rundt Dokumentprosjektet er det viktig å kjenne historikken og konteksten rundt.

4.3.1 Omorganisering av SITS

Skatteetaten gikk gjennom et prosjekt som het Reorganisering av Skatteetaten (ROS) i 2008. Den gang ble IT-funksjonen lite påvirket. IT-funksjonen, SITS, gikk derfor gjennom en omorganisering som ble innført 1. juni 2011. I denne omorganisering forsøkte man å ta høyde for moderniseringsløpet.

Seksjonen Felleskomponenter tar for seg det arkitekturmessige rundt integrasjonsplattform, systemutviklingsplattform og andre felleskomponenter. Andre seksjoner tar seg av det skattefaglige, innkreving, infrastruktur og grunndata. Omorganiseringen hadde fokus på funksjonelle perspektiver og systemansvar ligger innenfor hver av seksjonene.

Omorganiseringen førte til at mange ansatte fikk nye ansvarsområder og nye funksjoner. De påbegynte prosjektene fortsatte stort sett som før med tanke på bemanning.

4.3.2 Fornyelse av IT-funksjonen (FORIT)

Skatteetaten har et pågående program som heter «Fornyelse av IT-funksjonen». En del av dette programmet går på å modernisere IT-infrastrukturen. Dette er et program som er forankret helt opp til Skattedirektørens nivå, der han selv er leder for styringsgruppen. Fokuset for dette programmet er at Skatteetaten skal ha moderne og fleksible IT-løsninger. En del av dette vil være fokus på integrasjoner og tjenesteorientert arkitektur.

Man har ikke et eget SOA-program innen FORIT, men man har etablert nye prosesser som sier at Arkitekturenheten alltid skal være inne i prosjektene og styre hvordan de nye prosjektene skal utvikles. Målet er å få enhetlige integrasjoner som skal kjøres på en standardisert måte. Dette skal være kjent for alle som driver med systemutvikling i Skatteetaten.

Skatteetaten er langt fra ferdige med moderniseringsprogrammet. Det kjøres mange prosjekter parallelt som moderniserer forskjellige tjenester og funksjonalitet. Man uttaler at man skynder seg sakte og flytter bit for bit etter hvert som systemene har behov for fornying. Målet er gradvis å forenkle systemene sine ved å skille ut fellesfunksjonaliteter i felleskomponenter. Skatteetaten har enda langt igjen, og har en horisont på 5 til 10 år. De har laget rammeverkene for utviklingen og jobber kontinuerlig for å forbedre seg.

Skatteetaten har føringer som sier at all modernisering skal ha bakgrunn i et funksjonelt behov fra kunden. I tilfellet felles dokumentløsning så man at det ville være mest hensiktsmessig å løse kundens behov ved å utvikle en felleskomponent. Det var allerede flere systemer som hadde samme behov, og man visste at flere systemer ville få behovet etter hvert. Den gang prosjektet ble opprettet var det «Operativt råd for arkitektur og standarder» som tok avgjørelsen om å opprette en felleskomponent. I dag ville tilsvarende avgjørelse blitt tatt av «Arkitekturenheten».

Det finnes flere pågående prosjekter som har til hensikt å modernisere IT-funksjonen. Det jobbes blant annet med å modernisere Skatteetatens grunnlagsdata. Man har også et prosjekt som skal opprette elektroniske skattekort og dette skal tas videre i et prosjekt som sikrer elektronisk dialog mot arbeidsgivere for kontinuerlig å justere skattekort. Målet er å gjøre verden enklere for borgerne gjennom bedret informasjonsflyt og bedre tjenester.

4.3.3 I-plattform

Skatteetaten har 120 til 130 applikasjoner som er integrert ved hjelp av «spagetti-integrasjoner». Dette er direkte integrasjoner og hovedsakelig batch-integrasjoner. Disse integrasjonene er det ikke enkelt å vedlikeholde. Infrastrukturen var dessuten lite gjenbrukbar og komplisert bygget opp. Skatteetaten ønsket å få en felles integrasjonsplattform som integrasjonene kunne kjøres over. Det hadde vært gjort noen ustrukturerte forsøk med integrasjonsteknologi før man bestemte seg for å opprette et prosjekt som skulle innføre en felles integrasjonsplattform. Dette arbeidet startet rundt 2007-2008 og pågikk i omtrent halvannet år. Det ble da etablert rammeverk og retningslinjer for bruken som en første start. De senere årene har man jobbet videre med dette og fått et begrep om hvordan man skal gjøre integrasjoner i Skatteetaten.

Det man har erfart er at integrasjoner er mer enn en SOA-plattform. Skatteetaten må se sine integrasjoner i en helhet. I dette ligger det at det ikke er slik at alle integrasjoner nødvendigvis må gå over en integrasjonsplattform, men at man må integrere på en enhetlig måte. Det vil noen ganger være slik at en direkteintegrasjon er bedre enn å gå via en plattform, men man må være bevisst på når man direkte-integrerer og når man ikke gjør det. Det finnes i dag retningslinjer for hvordan man skal utvikle mot I-plattform og hvordan dette skal bygges opp arkitekturmessig. Alle som skal benytte I-plattform skal først få opplæring i hvordan dette gjøres. Det finnes ikke et sertifiseringsløp for utvikling mot I-plattform, men dette har vært diskutert.

Den siste tiden har Skatteetaten jobbet med sikring av komponentene i integrasjonene. Man har til nå stort sett erstattet koblinger en-til-en uten at man lager tjenester som gjenbrukes. Det man likevel oppnår er at de nye koblingene blir online-integrasjoner over en kontrollert plattform.

I tiden fremover vil det dukke opp flere felleskomponenter som ligger bak integrasjonsplattformen og tilbys ut til brukertjenester. Skatteetaten har bygget erfaring på bruk av integrasjonsplattformen og har bygget opp infrastruktur slik at alt går gjennom samme driftsmønster. Antall tjenester øker nå betydelig.

Flere og flere tjenester er nå oppdatert mot hverandre med online informasjonsflyt. Skatteetaten får også stadig flere integrasjoner mot eksterne samarbeidspartnere som domstoler og kemnere. Finansdepartementet har lagt føringer for hvordan de vil at offentlige systemer skal samhandle.

Skatteetaten utvikler systemer som tilfredsstill disse kravene, og integrasjonsplattformen er veldig viktig for sikkerheten i disse eksterne integrasjonene. Det bygges stadig flere tjenester som skal tilbys på portalen Skatteetaten.no.

4.3.4 NUPAS

Prosjektet NUPAS utviklet Aurora systemutviklingsplattform. Dette er en felles systemutviklingsplattform som er basert på Java, tjenesteorientering og nettleser. Målet med plattformen er at man skal få et felles standard utviklingsmiljø med mer effektiv kunnskapsdeling, systemutvikling og forvaltning av systemer.

Det er opprettet et sertifiseringsløp som alle som skal utvikle på den nye systemutviklingsplattformen må gjennomføre. Dette sertifiseringsløpet bygger på retningslinjer for hvordan man skal gjøre systemutvikling på den nye plattformen.

Retningslinjene er blitt utviklet parallelt med at Dokumentprosjektet har blitt gjennomført.

4.4 Fagsystemene i Dokumentprosjektet

Her beskrives de ulike fagsystemene som er involvert i Dokumentprosjektet. Dette vil gi leseren en bedre anledning til å forstå de enkelte fagsystemenes kapasiteter og behov.

4.5.2 Sofie

Sofie er saksbehandlings- og regnskapssystem for Skatteoppkreverne. Dette systemet benyttes av om lag 5000 saksbehandlere på 370 skatteoppkrevingkontor i hele landet. Systemet ble produksjonssett i 2005. Systemet driftes og utvikles av SITS sin avdeling i Oslo.

Sofie har om lag 100 brevmaler, og disse skrives ut i meget store volum.

I 2009 innså man at dokumentløsningen i Sofie var i ferd med å bli utdatert. Man så også at dokumentløsningen var veldig kompleks og tidkrevende å vedlikeholde. Hvis man for eksempel skulle endre en felleskomponent i et brev, som tittel linje eller endre plasseringen av et felt, så måtte man inn i alle de 100 brevmalene og gjøre endringene i hver av dem. Dette førte til at man i flere tilfeller valgte ikke å gjøre endringer som det egentlig var behov for fordi det ville kreve såpass mye arbeid å gjøre det. Det var mange feil i dokumentløsningen og arkitekturen var kompleks. Den tekniske løsningen og verktøyene bak den tidligere dokumentløsningen var utdaterte og dyre å oppdatere.

Det fantes ikke noen felles visuell og språklig profil i Skatteetaten. Det var ikke noe ved dokumentene som var gjenkjennbart og fortalte mottaker at dette var et brev fra Skatteetaten. Språkbruk fra dokument til dokument kunne være veldig forskjellig. Tjenesteeier var ansvarlig for utforming av nye dokumenter, og det var sjelden man fokuserte på å tilnærme utformingen til hvordan eksisterende dokumenter var lagt opp. En ny språklig og visuell profil for Skatteetaten er nå utviklet og man ønsker å gå fra et sterkt fokus på det juridiske innholdet til å la innholdet være mer informativt for mottaker. Den tidligere dokumentløsningen ga ikke saksbehandlerne anledning til å utforme brevene slik de

ønsket før endelig utsending. Fagsystemet nye dokumentløsning utvikles gjennom prosjektet SDOK som er et delprosjekt i Dokumentprosjektet.

Steria drifter Sofies produksjonsmiljø. Dokumentløsningen blir nå flyttet ut av Sofie. Dette kan være et argument og en besparelse ved reforhandling av driftsavtalen.

4.5.3 MVA

MVA er fagsystemet til faggruppen Avgift. Saksbehandlerne sitter i på de ulike skattekontorene, mens systemutvikling og drift gjøres fra SITS sin avdeling i Grimstad.

MVA har 550 ulike brevmaler i fagsystemet. Disse skrives ut og sendes til mottakere i små volum for hvert enkelt brev. I MVA konverterer man nå de eksisterende brevene fra den gamle brevløsningen til den nye brevløsningen. Dette er en stor jobb.

Det er en del forskjeller mellom den nye og den gamle brevløsningen. Den gamle var html-basert og hadde en dårlig editor. Løsningen var bundet til Internet Explorer 6. Dokumentløsningen var i tillegg veldig kompleks og tidkrevende å vedlikeholde. De gamle brevene hadde et tungvint, juridisk språk og hadde ikke et enhetlig utseende eller enhetlig ordlyd. Det ble bestemt at Skatteetaten skulle fornye sin språkprofil på samtlige brev og firmaet Arkitekst ble leid inn for å fornye brevene. De hadde derfor et behov for å fornye dokumentløsningen. Dette ble gjort gjennom prosjektet MDOK som var en del av Dokumentprosjektet.

Man har i MVA opplevd motstand fra saksbehandlerne mot de nye brevmalene. Man opplever at de oppretter brevet etter den nye malen. Deretter editerer de brevet eller kopierer innholdet fra den gamle brevmalen før det sendes ut til mottaker. Dette fører til at mottakerne mottar brev som er formatterte etter den gamle språkprofilen. Det kjøres i dag et løp der man informerer saksbehandlerne om at man må benytte de nye brevmalene.

4.5.4 Print

Print-miljøet holder til i Oslo og håndterer sentral utskrift for Skatteetaten og flere andre offentlige institusjoner med store utskriftsbehov.

Tidligere fikk Printmiljøet printfiler som var formattert på mange forskjellige måter fra sine kunder. Hver fil måtte da omformateres, og det oppsto ofte feil.

En standardisert felles innkanal vil gi printmiljøet ett felles grensesnitt. Dette gir dem også mulighet til å slå sammen printjobber slik at man kan drive mer effektivt.

Printmiljøet utvikler sin komponent i Dokumentprosjektet gjennom delprosjektet SporDok.

5. Forskningstilnærming

I dette kapitlet forklares det hvilken forskningstilnærming som har vært benyttet i studien. Først forklares studiens forskningsperspektiv, deretter beskrives det hvordan undersøkelsen er utformet og gjennomført, studiens metodiske tilnærming til problemstillingen og tilhørende dataanalyse. Til slutt vurderes forskningens kvalitet og hvilke muligheter som finnes for å generalisere basert på funn fra studien.

5.1 Forskningsperspektiv og filosofiske betraktninger

Forskningsperspektivet beskriver den måten en forsker ser virkeligheten på. Dette vil også påvirke hvordan en forsker analyserer data og genererer kunnskap.

Ontologi og epistemologi er to viktige begreper som forskeren må ha et avklart forhold til. Ontologi handler om på hvilken måte forskeren betrakter virkeligheten på og hvordan denne virkeligheten skapes, mens epistemologi handler om hvordan kunnskap genereres. Innen ontologi skiller man mellom ekstern realisme, intern realisme og subjektiv idealisme illustrert i tabell 3 (Walsham, 1995). Innen epistemologien skiller man mellom positivistisk, kritisk og fortolkende perspektiver på hvordan kunnskap genereres oppsummert i tabell 4 (Klein & Myers, 1999). Walsham (1995) benytter litt andre begreper og deler inn i positivisme, ikke-positivisme og normativisme.

Tabell 3 Ontologi (Walsham, 1995)

Begrep:	Beskrivelse:
Ekstern realisme	Virkeligheten eksisterer uavhengig av hvordan vi tolker den
Intern realisme	Virkeligheten er subjektiv og eksisterer i den kognitive forståelsen mellom mennesker
Subjektiv idealisme	Hver enkelt person konstruerer sin egen virkelighet

Tabell 4 Epistemologi (Klein & Myers, 1999; Walsham, 1995)

Begrep:	Beskrivelse:
Positivistisk	Positivistiske studier kjennetegnes av at de kommer med formelle anbefalinger, bruk av målbare variabler, testing av hypoteser eller at man trekker slutninger fra et representativt utvalg (generalisering). Akademisk kunnskap kan kun hentes ut fra fakta.
Fortolkende	Fortolkende studier kjennetegnes ved at man antar at kunnskap genereres ved hjelp av sosiale konstruksjoner som språk, bevissthet, delte oppfatninger, dokumenter, verktøy og andre artefakter.
Kritisk	Kritiske studier kjennetegnes ved at man kommer med sosial kritikk av eksisterende teorier og sannheter for å bekjempe status quo.

Som forsker oppfatter jeg virkeligheten gjennom subjektiv idealisme. Jeg tror at hver enkelt person konstruerer sin versjon av virkeligheten gjennom sine tidligere erfaringer og kunnskap. Dette betyr at måten jeg fortolker virkeligheten på i mine studier vil påvirkes av den kunnskap og de erfaringer som jeg har tilegnet meg gjennom livet, og denne virkeligheten vil nødvendigvis ikke tolkes på tilsvarende måte av en annen forsker.

Min forskningsfilosofi er at jeg gjennom et fortolkende syn mener at kunnskap genereres i samspillet mellom forskeren og deltakerne. Sammen med en subjektiv idealisme mener jeg at jeg som forsker fortolker en virkelighet basert på kunnskapen som genereres gjennom interaksjon mellom meg som forsker og mine deltakere. Det er viktig for meg som forsker å være klar over at jeg har dette perspektivet på forskningen fordi dette vil påvirke måten jeg legger opp forskningsstrategien og måten jeg analyserer og generaliserer dataene på.

5.2 Forskningsdesign

Med denne studien ønsker jeg å undersøke hvilke faktorer som påvirker gjennomføringen av et konkret systemutviklingsprosjekt som innfører en felleskomponent i en tjenesteorientert arkitektur. Hensikten var å finne relevant litteratur om hvordan man kan styre og gjennomføre et SOA systemutviklingsprosjekt ved bruk av smidig metode i et komplekst miljø. Et litteratursøk på søkeord som «SOA governance», «SOA project» og «SOA complexity» viste at det var lite relevant forskning på konkrete systemutviklingsprosjekter. Dette inntrykket bekreftes av Joachims (2011) litteratursøk og studie av forskning relatert til SOA. Denne delen av studien vil derfor dels være teoribyggende. Jeg begynte med rammeverkene utviklet av systemleverandører (BEA, 2005; Oracle, 2007; Oracle, 2010) og et fokus på SOA governance, men var bevisst på å benytte teorien som en del av en iterativ prosess mellom innsamling av data og analyse (Eisenhardt, 1989). Walsham (1995) sier at *«det er ønskelig i fortolkende studier å beholde en betydelig grad av åpenhet overfor dataene og være villig til å modifisere sine grunnleggende antakelser og teorier. Dette leder til en iterativ prosess mellom datainnsamling og analyse der de teorier man hadde innledningsvis utvides, revideres eller forkastes»*.

Jeg tok utgangspunkt i to rammeverk utviklet av systemleverandører (BEA, 2005; Oracle, 2010) som beskriver viktige fokusområder ved innføring og styring av et SOA-program. Disse rammeverkene fokuserer på hele organisasjonens innsats i et fullstendig SOA-program. De beskriver godt helheten i organisasjonens tilnærming til SOA governance, og følgene av denne helhetlige satsingen vil nødvendigvis være synlige i det enkelte prosjekt.

Datainnsamling og analyse har vært utført som iterative prosesser, og uventede funn har resultert i endring i fokusområdene underveis. Et sosio-teknisk perspektiv har vært sentralt for å tolke resultatene. I tillegg har jeg benyttet forskningslitteratur som fokuserer på SOA generelt, IT-arkitektur og kompleksitet, og smidig systemutvikling.

5.2.1 Valg av case

I denne studien valgte jeg å benytte en fortolkningsbasert singel casestudie for å analysere et konkret SOA-prosjekt som omfatter innføring av fellestjenester i Skatteetaten. Benbasat (1987) argumenterer for enkeltstående case-studier er gunstige hvis studien er av utforskende karakter. Jeg ønsket å forstå hvordan SOA-prosjekter gjennomføres i praksis og hvilke faktorer som påvirker utfallet av denne type prosjekt. Skatteetaten har i noen år vært i gang med en SOA-innføring og var derfor en aktuell forskningssetting for denne problemstillingen. UiA hadde kontakter i Skatteetaten, og tilgang til etaten ble derfor mulig.

Skatteetaten er en stor organisasjon som startet sitt SOA-program i 2009. De har en kompleks IT-infrastruktur som de er i gang med å fornye. Denne moderniseringen har omfattet innføring av en integrasjonsplattform og en felles systemutviklingsplattform. De er nå i gang med å gjennomføre flere prosjekter hvor de høster tjenester og moderniserer infrastrukturen litt etter litt.

Dokumentprosjektet er et sentralt prosjekt i forbindelse med fornyelsen av IT-infrastrukturen. Prosjektet skal innføre en fellestjeneste som fungerer som et grensesnitt mot en felles utskriftsløsning (heretter kalt «printløsning»). Samtidig skal man utvikle brukertjenester i to av fagsystemene som skal benytte denne fellestjenesten for å dekke sine utskriftsbehov. Prosjektet er stort og involverer mange interessenter på tvers av Skatteetaten og flere fysiske lokasjoner.

5.3 Forskningsstrategi og metodisk tilnærming

I denne studien har jeg benyttet en kvalitativ tilnærming, og casestudie har vært min forskningsmetode. Med et utforskende forskningsspørsmål så vil en kvalitativ tilnærming basert på analyse av én casestudie kunne gi nyttig informasjon. En kvalitativ analyse kan ifølge Eisenhardt (1989) være passende når forskningsområdet er relativt nytt eller man ønsker å øke kunnskapen om et område ved å komme frem til nye teorier.

Benbasat (1987) oppgir tre årsaker til at casestudier kan gi gode resultater når man forsker på informasjonssystemer:

1. Forskeren kan studere informasjonssystemene i sin naturlige setting, lære av det og generere nye teorier fra praksis
2. En casestudie lar forskeren svare på «hvordan» og «hvorfor» spørsmål og bidrar til å forstå bakgrunnen for kompleksiteten i prosessene som skjer underveis
3. En casestudie passer bra når man forsker på områder hvor tidligere forskning er mangelfull

I denne studien passet alle de tre grunnene, og jeg følte at en casestudie var en metode som passet til å besvare mine forskningsspørsmål.

Walsham (1995) har beskrevet viktige momenter som man bør ta hensyn til ved gjennomføring av empiriske casestudier. Han sier at forskeren må ha et klart forhold til sin egen rolle i prosessen, det fenomenet han skal studere, og deltakerne i undersøkelsen. Forskeren skal analysere og rapportere de funnene han gjør, men forskerens rolle vil påvirke deltakerne i noen grad. For meg var det naturlig å innta rollen som observatør på utsiden av prosjektorganisasjonen. Jeg hadde ingen tilknytning til Skatteetaten og direkte deltakelse i prosjektet var ikke et mulig alternativ. En rolle som observatør har både fordeler og ulemper (ibid).

Jeg mener at ved å opptre som observatør vil jeg påvirke prosjektet i minst mulig grad, og få data som er minst mulig påvirket av mine holdninger. Ved å komme fra utsiden vil jeg også kunne oppleves som «nøytral» og kunne få ærligere tilbakemeldinger fra de involverte. Dette forutsetter at jeg klarer å skape et tillitsforhold mellom meg som forsker og de involverte i prosjektet (Benbasat, 1987). Rollen som observatør kan ha noen ulemper, som at jeg ikke nødvendigvis er til stede når hendelsene inntreffer (Walsham, 1995), men jeg mener at jeg vil kunne dekke dette ved å være bevisst utfordringene og hente inn data fra flere ulike kilder, triangulere (Jakobsen, 2005), og analysere disse for få et riktig bilde av helheten.

Jeg har benyttet kvalitative intervjuer i studien og Myers & Newman (2007) og Walsham (1995, 2006) har vært sentrale artikler da de gir retningslinjer for hvordan intervjuer skal gjennomføres og hvordan forskeren bør opptre.

Walsham (1995) uttaler at *«viktigheten av sosiale faktorer relatert til informasjonssystemer har blitt stadig mer anerkjent, og dette har ført til at IS-forskere i større grad benytter empiriske tilnærminger som fokuserer på menneskenes tolkninger og meninger»*. I min studie har jeg kombinert et sosio-teknisk perspektiv med en fortolkende tilnærming og gjennomført en casestudie hvor fokus har vært rettet mot innføringen av en felleskomponent i Skatteetatens SOA-program.

Studien har fulgt et konkret systemutviklingsprosjekt i Skatteetaten. Målet har vært å øke kunnskapen om hvilke sosio-tekniske faktorer som påvirker utviklingen av en felleskomponent i en kompleks tjenesteorientert arkitektur. Jeg håper videre å kunne komme med anbefalinger til hvordan prosjekter i framtiden bør ta høyde for de utfordringene som ble identifisert i denne studien.

5.3.1 Datainnsamling

Yin (1984) oppgir fem ulike kilder til informasjon når man gjennomfører en casestudie. Disse er:

1. Dokumentasjon
2. Arkiverte data
3. Intervjuer
4. Direkte observasjon
5. Fysiske artefakter

Det viktige er å skaffe seg grundige data angående forskningsobjektet og den komplekse konteksten som dataene eksisterer i. Hvilke datakilder man benytter vil avhenge av både forskningsspørsmålet og de resultatene man analyserer. Det er viktig å planlegge dette godt for å forsikre seg om at man oppnår tilstrekkelig datarikhet til å kunne besvare forskningsspørsmålet og for å benytte tiden effektivt (Benbasat, 1987).

Jeg har benyttet semistrukturerte intervjuer som primær datakilde og dokumenter som sekundær datakilde i denne studien. I tillegg har jeg hatt noen uformelle samtaler med en av prosjektlederne.

Valg av deltakere til studien

For å få best mulige resultater fra studien var det viktig å få deltakere som kunne gi informasjon fra ulike deler av prosjektprosessen slik at man fikk en god forståelse for prosjektet som helhet. Jeg intervjuet derfor alle prosjektlederne fra hvert delprosjekt, og intervjuet også personer i ulike prosjektroller slik at et bredt spekter av prosjektarbeidet ble dekket. Ved å intervju de tre delprosjektlederne mener jeg at jeg fikk inn viktig informasjon fra personer som har sittet helt sentralt gjennom hele prosjektløpet og som har kjennskap til bakgrunnen for mange av de avgjørelsene som er blitt tatt underveis. De resterende intervjuene har gitt meg anledning til å få et innblikk i hvordan resten av organisasjonen har opplevd disse beslutningene. Intervjuene ble gjennomført over ett år med deltakere som hadde følgende roller i prosjektet (Tabell 5): prosjektleder for FDOK, prosjektleder for MDOK, prosjektleder for SDOK, ett medlem fra styringsgruppen i hovedprosjektet, en systemutvikler i MDOK, en systemutvikler i FDOK, en

systemarkitekt som jobber med tjenesteorientert arkitektur i Skatteetaten, samt en systemutvikler som jobber på printsenteret.

Tabell 5 Deltakere i studien

Rolle:	Prosjekt:	Dato:
Prosjektleder for FDOK	FDOK	27.01.2012
Medlem i styringsgruppen	Dokumentprosjektet	03.02.2012
Systemarkitekt fra Arkitekturenheten	Dokumentprosjektet	17.02.2012
Systemutvikler i MDOK	MDOK	22.02.2012
Prosjektleder for SDOK	SDOK	23.02.2012
Prosjektleder for FDOK	FDOK	15.06.2012
Systemutvikler i FDOK	FDOK	27.09.2012
Prosjektleder for MDOK	MDOK	11.10.2012
Systemutvikler i SporDok	SporDok	11.10.2012

Gjennomføring av intervjuer

Intervjuene ble gjennomført som semistrukturerte intervjuer med ulike deltakere som alle på en eller annen måte var involvert i de tre delprosjektene som utgjør Dokumentprosjektet (Tabell 5).

Intervjuene ble gjennomført på ulike tidspunkter i prosjektet.

Intervjuene ble etter avtale med deltakerne tatt opp ved hjelp av taleopptaker da dette gir meg som forsker anledning til å fokusere på detaljer i intervjuet (Walsham, 2006). Analysen begynte med at jeg hørte gjennom intervjuene flere ganger, og det ble skrevet sammendrag av hvert intervju. Hvert intervju varte i omtrent 60 minutter.

Jeg prøvde å benytte åpne spørsmål så langt som mulig i intervjuene for å gi deltakerne anledning til å dele sine erfaringer uten at mine synspunkter og holdninger skulle påvirke svarene i den første fasen, mens oppfølgingsspørsmål naturlig nok i større grad var mer strukturerte og avgrensede til spesifikke tema. Jeg prøvde å være bevisst på og ikke ta med meg for mange erfaringer fra tidligere intervjuer inn i de nye intervjuene. Dette for å prøve å stille med så blanke ark som mulig og ha et åpent sinn i forhold til den informasjonen som ble gitt i intervjuet.

Intervjuene ble gjennomført på ulike tidspunkter. Det var flere årsaker til dette. Én årsak var den praktiske gjennomførbarheten i forhold til forskerens og deltakernes timeplan. Den viktigste årsaken var dog at siden dette er en fortolkende studie så har forskningsspørsmål og teorier modnet seg underveis. Jeg startet derfor med å intervjuer deltakere som hadde god oversikt over Skatteetatens prosjekter og systemutviklingspraksis for å hente inn informasjon bredt, mens intervjuene etter hvert ble mer avgrenset til spesifikke tema.

Ved å gjennomføre intervjuer over et litt lengre tidsrom fikk jeg sannsynligvis litt andre resultater enn hvis alle intervjuene hadde blitt gjennomført på et tidlig tidspunkt. Prosjektene utviklet seg mye i denne perioden, og jeg mener at mange av funnene kom nettopp fordi noen intervjuer ble gjennomført på et senere tidspunkt. Jeg har prøvd å triangulere funnene mine ved å intervjuer deltakere fra flere ulike deler av prosjektet for å fange inn ulike synspunkter, og ved å gjennomføre intervjuene til ulike tidspunkter for å fange opp hele utviklingen i prosjektet.

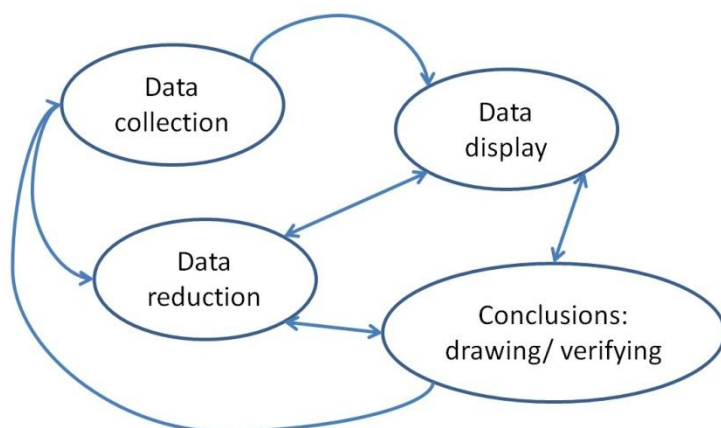
I noen tilfeller har jeg kontaktet kilder på nytt for å få mer utdypende informasjon om hendelser som ble nevnt i intervjuet.

Noen av aktørene i prosjektet har jeg ikke fått tilgang til å intervju. Den første prosjektlederen i FDOK hadde sluttet som konsulent i prosjektet, og jeg fikk derfor ikke anledning til å intervju ham. Det kan godt tenkes at han kunne hatt mye interessant informasjon om gangen i prosjektet. Jeg føler likevel at jeg har dekket dette informasjonsgapet ved å intervju mange deltakere som overlapper hverandre og som i sum gir det korrekte bildet av prosjektets gang.

Noe informasjon har jeg også fått tilgang til i form av prosjektdokumenter og Skatteetatens hjemmesider. Dette er benyttet som sekundærkilder i studien- og er derfor viktig støtteinformasjon til resultatene fra intervjuene.

5.3.2 Dataanalyse

I analysen av dataene mine har jeg fulgt Miles & Hubermans (1994) interaktive modell. De uttaler at analysen vil skje i et samspill mellom datainnsamling, datareduksjon, datavisning og konklusjon illustrert i figur 16.



Figur 16 Components of Data Analysis (Miles & Huberman, 1994)

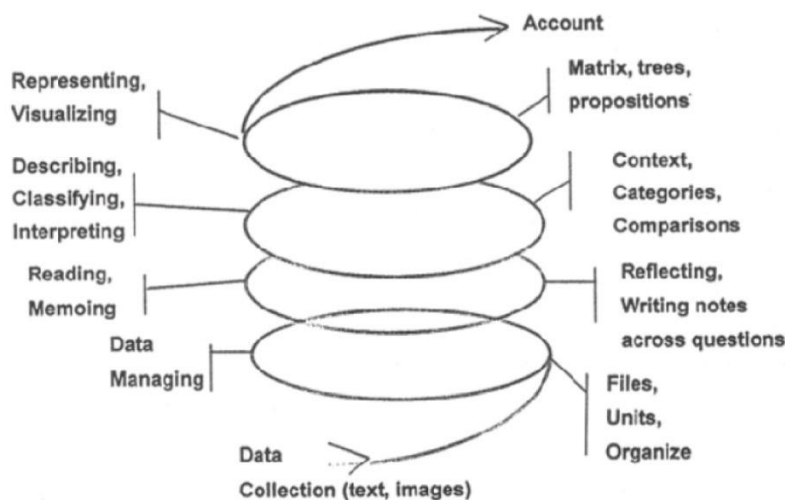
Datainnsamlingen ble primært gjort i form av semistrukturerte intervjuer som ble tatt opp, samt notater som ble gjort underveis i intervjuene.

Datareduksjon ble gjort ved å lage tankekart mellom prosjektene og samle inn de viktigste holdepunktene fra hvert intervju. Lydfilene ga meg en mulighet til å lytte til detaljene i hva deltakerne sa og i hvilken sammenheng informasjonen ble gitt, samtidig som notatene og sammendragene hjalp til ved å danne det store bildet. Etter hvert som intervjuene ble gjennomført begynte det å tegne seg et bilde som viste helheten i Dokumentprosjektet. Siden intervjuene ble gjennomført på ulike tidspunkter bidro denne reduksjonen til å gi innspill til den videre datainnsamlingen.

Visuelle bilder av viktige data var basert på tankekartene og notatene, men de ble også lagt inn som stikkord i masteroppgaven underveis. På denne måten kunne egne refleksjoner modnes og analysen basert på datareduksjon og ny datainnsamling fullføres over tid.

Konklusjonene ble utviklet gradvis utover i prosjektet og ble testet opp mot nye intervjuer og dokumenter. Kategoriene som utgjør funnene mine ble tydeligere etter hvert som intervjuene pekte på mange av de samme årsakene til forsinkelser og utfordringer. Mange av de sammen momentene som deltakerne nevnte stemmer overens med erfaringer man finner i litteratur som omhandler sosio-tekniske utfordringer i et komplekst systemutviklingsmiljø.

Prinsippene fra den hermeneutiske sirkel har lagt til grunn for min fortolkning og forståelse av dataene som har kommet ut av studien (Creswell, 1998; Klein & Myers, 1999). Ved å forstå de ulike delene av prosjektet og sammenhengene mellom disse har jeg kunnet sette dette sammen til et større bilde for å kunne si noe om helheten. Figur 17 viser den hermeneutiske sirkel og hvordan man går fra data til forståelse.



Figur 17. Den hermeneutiske sirkel (Creswell, 1998)

5.4 Forskningens kvalitet

Klein & Myers (1999) beskriver 7 prinsipper for å vurdere kvaliteten i et fortolkende studium.

Det fundamentale prinsippet av den hermeneutiske sirkel

Dette prinsippet sier at menneskelig forståelse bygges ved iterasjoner mellom å tolke sammenhengene mellom de enkelte bestanddelene og den helheten som de utgjør.

Prinsipp om kontekstualisering

Krever en kritisk refleksjon av den sosiale og historiske bakgrunnen for forskningssettingen, slik at leserne kan se hvordan den nåværende situasjonen oppsto.

Prinsipp om interaksjon mellom forsker og objektene

Krever en kritisk refleksjon av om dataene eksisterer eller ble til gjennom interaksjon mellom forskeren og deltakerne.

Prinsipp om abstraksjon og generalisering

Krever at man oversetter dataene til teoretiske og generelle konsepter

Prinsipp om tilbakevirkende meningsbygging (dialogical reasoning)

Krever at man er kritisk til egne funn ved å revidere dem i flere iterasjoner

Prinsipp om flere fortolkninger

Krever at man er følsom overfor forskjellige fortolkninger mellom deltakere

Prinsipp om mistenksomhet

Krever at man er mistenksom overfor deltakernes uttalelser og ikke tar dem som den fulle sannhet

5.4.1 Vurdering av forskningens kvalitet

Jeg har anvendt Klein & Myers (ibid) prinsipper for å vurdere kvaliteten i min studie. Dette ga meg en mulighet til å være bevisst på min rolle som forsker. Jeg har vært bevisst på de 7 prinsippene underveis i studien. Mine erfaringer fra dette er oppsummert i tabell 6.

Tabell 6 Forskningens kvalitet etter Klein & Myers (1999) prinsipper

Prinsipp:	Beskrivelse:
Fundamentale prinsipp om den hermeneutiske sirkel	Prinsippet om den hermeneutiske sirkel har ligget til grunn for gjennomføringen av denne studien. Det har vært en iterativ prosess der både datainnsamling og analyse har skjedd over tid. Jeg har prøvd å fortolke prosjektene og intervjuene for seg, for så å forstå det store bildet.
Prinsipp om kontekstualisering	Jeg har prøvd å forstå konteksten og historikken som prosjektet eksisterer i ved å beskrive både Skatteetaten som etat og konteksten de opererer i. Jeg har i tillegg prøvd å illustrere begreper rundt tjenesteorientering og IT-arkitektur som legger føringer for fremtidige valg.
Prinsipp om interaksjon mellom forsker og objektene	Jeg har vært bevisst Myers & Newmans (2007) anbefalinger og fallgruver under datainnsamling. Jeg har prøvd å skape en god atmosfære under intervjuene og gi deltakerne trygghet (Walsham, 2006). I løpet av intervjuene har både jeg og deltakerne gitt fortolkninger i samspill. Deltakerne har validert funn ved å diskutere disse med forskeren og omvendt. På denne måten har jeg benyttet muligheten til å få bekreftet funn ved å diskutere med deltakerne om funnene har vært forstått riktig (Bygstad og Munkvold, 2007). Utover dette har en av prosjektlederne mottatt studien for gjennomlesing for å gi tilbakemelding om eventuelle uklarheter.
Prinsipp om abstraksjon og generalisering	Ved å benytte en singel casestudie kan det være vanskelig å generalisere funnene. Dette er et utforskende studium der jeg søker å oppnå økt kunnskap om hvilke faktorer som påvirker et konkret SOA-prosjekt. Jeg håper at jeg ved å søke i litteraturen og sette sammen ulike synspunkter klarer å vise en sammenheng mellom mine data og mine konklusjoner. Jeg har forsøkt gjennom intervjuene å få inn rike nok data fra ulike deltakere slik at man kan

	se en helhet i prosjektet. Forhåpentligvis kan senere studier følge opp mine funn og bygge videre på disse.
Prinsipp om tilbakevirkende meningsbygging	Jeg har prøvd å være bevisst på å beholde et åpent sinn mellom ulike intervjuer for å kunne stille uten for mange inngangsverdier. Jeg har vært åpen for å modifisere grunnleggende antakelser underveis, basert på informasjonen fra deltakerne. Jeg har så prøvd å redusere og analysere hvert intervju for seg, før jeg har prøvd å se om funnene forteller det samme helhetlige bildet.
Prinsipp om flere fortolkninger	Jeg har prøvd å intervju personer fra ulike deler av prosjektet for å få med meg så mange fortolkninger som mulig. Når innspillene fra aktører i ulike deler av prosjektet peker i samme retning, så har dette bidratt til å kunne identifisere felles mønstre og gitt muligheter til å kategorisere funnene
Prinsipp om mistenksomhet	Det er vanskelig å si i hvilken grad jeg har klart å oppnå dette prinsippet. Jeg har prøvd å ikke konkludere eller ta noen avgjørelser basert på enkeltuttalelser, men stille meg kritisk til meninger som ikke har blitt verifisert. Det var et tilfelle hvor den ene deltakeren uttalte seg kritisk om en prosess i et av de andre prosjektene. Jeg følte da at jeg måtte forholde meg kritisk til uttalelsen for ikke å la den farge mitt syn. Ved et senere intervju ble denne uttalelsen bekreftet av et medlem i prosjektet som ble kritisert, og jeg følte da at jeg kunne benytte den originale kritikken i studien.

Walsham (1995) uttaler at selv om en slik analyse av forskningens kvalitet som er presentert i tabellen over er nyttig, så er det viktig ikke å la seg blende av en slik evaluering. Det er mulig å gjennomføre en studie etter Klein & Myers prinsipper uten at den leverer interessante resultater. Jeg mener at min studie gir økt kunnskap om utvikling av felleskomponenter i en tjenesteorientert arkitektur som kan benyttes av både forskere og praktikere i fremtiden. Så blir det opp til leseren å vurdere om han/hun er enig i dette.

5.4.2 Bruk av teori i studien

Jeg har hatt en utfordring med å finne teori som jeg føler passer i forhold til det bildet jeg så i Skatteetaten og de forskningsspørsmålene som jeg jobbet etter. Det synes å være få studier som ser på konkrete systemutviklingsprosjekter innen tjenesteorientert arkitektur og hvordan disse gjennomføres (Joachim, 2011). Det kan synes som at mange mener at systemutvikling er systemutvikling uansett hvilken kontekst det gjennomføres innen. Siden jeg ikke har funnet så mye forskning på konkret systemutvikling innen tjenesteorientert arkitektur så har jeg støttet meg til litteratur fra gjennomføring av systemutvikling ved hjelp av smidige metoder, da det er dette som ble benyttet i Skatteetaten.

5.4.3 Min rolle som forsker

Min rolle i Dokumentprosjektet i Skatteetaten har vært at jeg har vært ekstern. Jeg har ikke hatt noen direkte tilknytning til Skatteetaten og har heller ikke deltatt i selve prosjektet. Dette kan gi både

fordeler og ulemper. En fordel er at jeg som ekstern kan se på prosjektet med nye øyner uten å være farget av hverken etaten eller tidligere prosjekter. Jeg kan derfor fritt komme til mine egne konklusjoner basert på de observasjonene som jeg gjør i prosjektet.

En ulempe er at jeg ikke kjenner Skatteetaten i detalj og at jeg heller ikke har sittet i prosjektet hele veien. Skatteetaten er en stor og kompleks organisasjon og det synes å pågå en kraftig modernisering av IT-infrastrukturen. Dette fører til at det er mange samtidige prosjekter som pågår og mange av disse vil påvirke hverandre. Som ekstern har jeg opplevd det som vanskelig å sette meg inn i hvordan Skatteetaten jobber og hvordan de ulike prosjektene hører sammen i porteføljen, og det har krevd mye innsats å få en oversikt over dette. Videre så har det vært litt vanskeligere å komme utenfra og prøve å holde seg a jour i prosjektene når jeg ikke deltar i prosjektmøter og jobber med problemstillingene daglig.

Jeg mener likevel at jeg har skaffet meg en god oversikt og fått et godt overblikk over Dokumentprosjektets utvikling i Skatteetaten. Jeg føler jeg har klart å samle den nødvendige informasjonen for å kunne danne meg et helhetlig bilde av hva som har fungert godt og hva som kunne vært gjort annerledes i Dokumentprosjektet.

6. Resultater og analyse

I dette kapitlet presenteres de viktigste resultatene fra studien og det gjøres en analyse av disse. Studien har fulgt deler av prosessen til Dokumentprosjektet og dets fire delprosjekter i Skatteetaten som har gått parallelt i perioden fra 2009 til 2012. Prosjektet skal avsluttes i januar 2013.

Prosjektene er viktige implementeringsprosjekter og utgjør elementer i strategien for å oppnå en tjenesteorientert arkitektur i etaten. FDOK omtales som den første virkelige felleskomponenten i Skatteetaten, og man håper at dette kan være et signalprosjekt som kan vise hvilke muligheter en tjenesteorientert arkitektur kan gi.

6.1 Dagens prosjekter og status

Her beskrives sentrale aktiviteter i ulike prosjekter som inngår i Dokumentprosjektet. Det er her snakk om fire prosjekter som har vært utført parallelt. Kapitlet vil først beskrive hele prosjektet under ett for å gi en forståelse av helheten og den generelle flyten i de prosjektene som inngår. Deretter vil hvert av delprosjektene og den flyten som har vært internt i hvert av dem beskrives. Avslutningsvis beskrives hvilke potensielle effekter Dokumentprosjektet har som formål å oppnå og spesielle utfordringer i prosjektet.

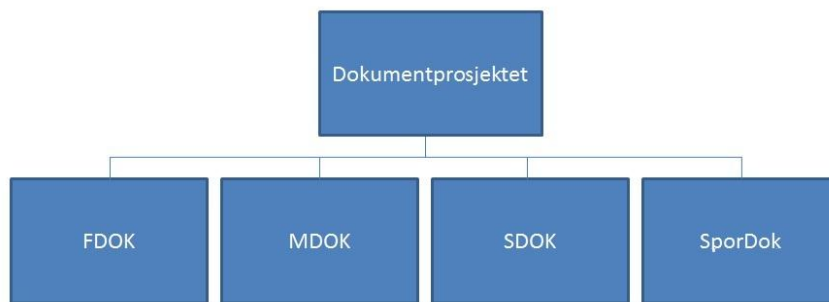
Opprinnelig startet Dokumentprosjektet som et initiativ fra ulike fagavdelinger som hadde hvert sitt fagsystem og hvor man ønsket å modernisere sine brevløsninger i sine respektive siloer. Det viste seg at de ulike fagmiljøene hadde de samme utfordringene. Man kom frem til at initiativene relatert til brevløsninger burde samles slik at man opprettet én felleskomponent som tilbød et felles grensesnitt mot komponenter i hvert enkelt fagsystem nedover i verdikjeden. Samtidig skulle felleskomponenten tilby ett grensesnitt mot Print-miljøet. Prosjektene FDOK (Felleskomponent), MDOK (MVAs komponent) og SDOK (Skatteoppkrevernes fagsystem, Sofies, komponent), og senere SporDok i Print-miljøet, ble så opprettet (se figur 18).



Figur 18 Dokumentprosjektets organisering i første fase

Det fantes i starten ikke noen formell overbygging over disse prosjektene, men prosjektlederne og sentrale aktører i Skatteetaten sørget for koordinering mellom dem. Mot slutten av 2011 ble det opprettet en formell overbygging kalt Dokumentprosjektet, der FDOK, MDOK, SDOK og SporDok gikk inn som delprosjekter (se figur 19). I studien benyttes begrepet Dokumentprosjektet de gangene hele

brevløsningen omtales, også i perioden før Dokumentprosjektet offisielt startet opp.



Figur 19 Dokumentprosjektets organisering i siste fase

6.1.1 Dokumentprosjektet

Dokumentprosjektet ble initiert fra MVA-avdelingen i 2009 som da hadde utviklet et sterkt behov for en ny brevløsning. Den eksisterende løsningen var bundet til Internet Explorer 6 og hadde en dårlig editor i tillegg til flere andre mangler. Den var også kostbar å vedlikeholde. MVA-avdelingen startet derfor å undersøke muligheter for hvordan de kunne modernisere brevløsningen innad i deres silo. I en uformell sammenheng i Skatteetaten ble dette diskutert. Det ble da nevnt at MVA-avdelingen jobbet med å forbedre sin brevløsning. Det viste at Sofie hadde de samme utfordringene med sin kjøpte brevløsning. I Sofie hadde de ikke mulighet til å endre sine eksisterende brev og det var derfor knyttet store kostnader til drift og oppdateringer av løsningen. Man visste at dette behovet også eksisterte i flere andre fagavdelinger i Skatteetaten som hadde tilsvarende fagsystemer og brevløsninger. Det var derfor et behov for å utvikle en felles tjeneste (felleskomponent) som kunne inngå i en tjenesteorientert struktur som ville dekke hele Skatteetatens behov.

Det ble sett på muligheten for å anskaffe en hyllevareløsning som var en ferdig dokumentløsning klar til bruk. Parallelt med dette vurderte man også andre alternativer og man kjørte et lite «Proof of Concept»-prosjekt i MVA der man programmerte en enkel løsning in-house for å se på hvilke muligheter man hadde til å utvikle en tjeneste lokalt. Denne POCen ble presentert internt i Skatteetaten. De vurderte også eksisterende dokumentløsninger i andre organisasjoner som NAV, Lånkassen, Mattilsynet og BBS, og noen kommersielle leverandører. De store verktøyene hadde mye funksjonalitet og kunne gjøre mye dokumentproduksjon, men man kom frem til at Skatteetaten kun hadde behov for dokumenter i Word- og pdf-format. De store løsningene var også av eldre utgave og var ikke egnet for tjenesteorientering. Man valgte derfor å anskaffe produktet som genererer dokumentene som hyllevare, og utvikle resten av løsningen internt.

Prosjektene FDOK, MDOK og SDOK ble derfor opprettet for å utvikle en helhetlig løsning. De tre prosjektene skulle utvikle de tjenestene som i første omgang skulle gi MVA og Sofie en moderne brevløsning. Tanken var at flere systemer skulle benytte fellestjenesten FDOK for sine dokumentbehov. Dette ble ansett som gunstig og en av deltakerne i undersøkelsen uttalte: *«når man får stadig flere systemer til å knytte seg opp mot felleskomponenten så begynner man virkelig å få effekt av prosjektet»*.

Man valgte opprinnelig å opprette tre frittstående prosjekter på grunn av finansieringen. FDOK og SDOK ble opprettet som egne prosjekter med øremerkede midler fra Finansdepartementet, mens

MDOK ble finansiert gjennom MVAs driftsmidler. I prosjektet FDOK hadde man to team, der ett satt i Oslo og ett i Grimstad. MDOK var samlet i Grimstad, mens SDOK var samlet i Oslo.

Dokumentprosjektet beskrives som den første store tjenesteorienterte felleskomponenten med et funksjonelt perspektiv som er produsert i Skatteetaten. Det er sånn sett et viktig prosjekt i deres SOA-program. Dette er et prosjekt som kan vise hvilke muligheter som ligger i felleskomponenter og blir sett på som en gunstig kandidat for å oppnå tjenesteorientering. FDOK inngår som ett av flere trinn for å få en felleskomponent rundt dokumenthåndtering. Her inngår dokumentproduksjon, journalføring, arkivering, korrespondanselogg og kanalvalg inn og ut.

Det er viktig at en felleskomponent legges på et riktig nivå i forhold til hva som er felles og hva som er fagspesifikt. Her var Dokumentprosjektet veldig lett å avgrense med tanke på hvilken funksjonalitet som hører hjemme i en felleskomponent og hvilken funksjonalitet som hører hjemme i fagsystemene.

Dokumentprosjektet møtte en del utfordringer fordi det var blant de første prosjektene som ble iverksatt som en del av den tjenesteorienterte strategien. De høstet derfor erfaringer som andre prosjekter kan dra nytte av hvis tilsvarende utfordringer skulle dukke opp. Dokumentprosjektet var også det første prosjektet som ble kjørt på den nye systemutviklingsplattformen, Aurora. Prosjektet ble delvis utført i forkant av NUPAS-prosjektet, som er ment å legge føringer for hvordan man skal utvikle systemer på den nye Java-plattformen. Dokumentprosjektet benytter seg også av I-plattform som er en tjeneste som har ansvar for all integrasjon mellom eksterne og interne systemer i Skatteetaten. Tidligere var fagavdelingene selv ansvarlige for sine fagsystemer og integrasjon mot andre systemer.

Deltakere i MVA-prosjektet foreslo at prosjektleder fra POC skulle fortsette som prosjektleder i FDOK. Dette ønsket ikke Sofie-miljøet og det ble derfor ansatt en ekstern prosjektleder i FDOK som var lokalisert i Oslo. Det ble leid inn flere eksterne Java-konsulenter til Oslo-kontoret fordi Skatteetaten hadde liten kompetanse og erfaring med systemutvikling i Java. Prosjektleder og konsulentene restrukturerte FDOK ut fra POCen, men involverte i liten grad fagmiljøet i Grimstad. Fagmiljøet i Grimstad ble derfor holdt utenfor, noe som medførte at samarbeidsklimaet ble dårlig. Fagmiljøet i Grimstad hadde laget POCen og følte de burde vært mer involvert i det videre arbeidet.

Tanken var at FDOK skulle presentere et felles grensesnitt mot Printmiljøet for leveranse av printdata. Det ble utarbeidet en kravspesifikasjon for FDOK. Dette var et omfattende arbeid hvor man søkte å involvere alle parter. Arkitekturovervåkeren var også med i noen av disse møtene. Man ble enige om kravene, som blant annet skulle legge til rette for flere utkanaler for dokumentene. Printmiljøet uttalte i intervjuer at de følte at de ble involvert for sent i dette arbeidet siden de først ble med etter at mange av føringene var lagt. Printmiljøet mente at det var en dårlig løsning å skulle utvikle dokumentløsningen selv, og mente at man heller burde anskaffe et større «document composition tool». De fikk ikke gjennomslag for dette synspunktet.

I starten av prosjektet hadde man møter på tvers av prosjektene ved behov (ca. hver annen uke) for å koordinere fremdriften. Når man var ferdig med kravspesifikasjonen og kontraktene mellom prosjektene ble disse møtene mye sjeldnere. Møtene tok imidlertid seg opp igjen når ulike utfordringer oppstod.

Kravspesifikasjonene og avtalene mellom de ulike prosjektene skulle være klare slik at systemene kunne begynne å utvikle sine tjenester. Man så etter hvert at oppdelingen i tre prosjekter ikke var heldig. Det var tre frittstående prosjekter der FDOKs fremdrift og valg av løsninger i for liten grad ble kommunisert til systemutviklerne i prosjektene i fagsystemene. I FDOKs kravspesifikasjon beskrev man hvordan grensesnittene mot fagsystemene skulle være. Disse grensesnittene skulle brukes som kontrakter mot FDOK. Prosjektene FDOK, MDOK, SDOK og SporDok startet samtidig og fagsystemene forutsatte at kravspesifikasjonen til grensesnittene skulle oppfylles. Systemutviklingen i prosjektene pågikk parallelt, og man var avhengig av at kontraktene ble fulgt. Hvis ett av prosjektene ikke utviklet i henhold til avtalene ville dette føre til utfordringer når de andre prosjektene utviklet sine grensesnitt som om avtalen ble fulgt. FDOK fikk tidlig opp sine viktigste tjenester slik at fagsystemenes prosjekter kunne teste mot disse. I starten så det ut til at disse tjenestene var utviklet i henhold til kontrakten. SDOK og MDOK har uttalt at de gikk ut i fra at FDOK ville utvikle resten av fellestjenesten slik man hadde avtalt.

Systemutviklingsprosjektene SDOK og MDOK fikk en del utfordringer på grunn av krav fra sentrale enheter i Skatteetaten. Personell i FDOK gjennomførte møter med personell fra Arkitektur, Sikkerhet, NUPAS og andre avdelinger i Skatteetaten uten å involvere representanter fra de andre prosjektene. Dette førte til at man avtalte eksempelvis sikkerhetskrav som fagsystemene ikke hadde mulighet til å tilfredstille. Personell fra prosjektene MDOK og SDOK måtte da gjennom en tung prosess og starte nye forhandlinger med Sikkerhet for å bli enige om hvordan man kunne komme frem til en sikker løsning som det var mulig å implementere. Representanter fra SDOK og MDOK uttalte at det var en klar bom fra FDOKs side å ikke involvere dem i de første møtene.

Når det nærmet seg høsten 2011 og tiden hvor FDOK skulle være leveringsklart ble man klar over at det var en del mangler i tjenesten. FDOK ville ikke klare å levere den funksjonaliteten som var forutsatt i kravspesifikasjonen og funksjonalitet mot utkanal var ikke testet i det hele tatt.

Fagsystemene ble klar over manglene på et meget sent tidspunkt, slik at det var veldig vanskelig å få gjort noe med disse problemene. Det var ikke penger igjen i prosjektet til å gjøre de nødvendige endringene som trengtes.

Prosjektleder i MDOK uttaler at hun og prosjektleder FDOK følte et massivt press ovenfra i Skatteetaten i denne perioden. Man fikk stadige spørsmål om når dokumentløsningen ville produksjonssettes og det ble påpekt at prosjektet var over tidsplanen. Hun uttaler at *«hadde det vært opp til både meg og prosjektleder for FDOK så hadde aldri FDOK og MDOK vært produksjonssatt i november 2011»*. Det som ble produksjonssatt i november 2011 var en minimumsløsning som man visste hadde flere kjente feil, men som ble produksjonssatt på grunn av presset. På spørsmål om hvor dette presset kom fra så svarer man at det kom fra nærmeste sjef og fra ledelsen i SITS, men at man ble fortalt at det også ble presset på helt fra Skattedirektøren.

Det ble foretatt en evaluering av prosjektene på høsten 2011 og det ble bestemt av FDOK skulle videreføres, men at man trengte en overbygning over prosjektene FDOK, MDOK, SDOK og SporDok. Dokumentprosjektet ble derfor en overbygging for disse tre prosjektene, og har som oppgave å koordinere innsatsen mellom de fire prosjektene, i tillegg til I-plattform. Det ble leid inn en ekstern konsulent til prosjektlederrollen for Dokumentprosjektet. Denne konsulenten hadde lang erfaring fra tidligere konsulenttjenester i Skatteetaten og hadde jobbet i systemutviklingsmiljøet relatert til Sofie. Dette skjedde omtrent samtidig med at FDOK ble overlevert til linjen og fikk en intern prosjektleder.

Flere av deltakerne i studien hevdet at de merket forbedringer i informasjonsflyt og koordinering etter at Dokumentprosjektet ble opprettet. Prosjektleder for Dokumentprosjektet har jobbet med å sikre informasjonsflyt og har økt hyppigheten av møter mellom de sentrale aktørene i de ulike prosjektene. I intensive perioder med høyt aktivitetsnivå har man daglige Scrum-møter hvor alle involverte parter deltar, og man søker å avdekke alle avhengigheter som kan påvirke resultatet.

Den nye prosjektlederen i FDOK gjennomførte en fullstendig gjennomgang av kravspesifikasjoner og estimerer da hun tok over ansvaret for prosjektet. Hun forteller at det raskt viste seg at systemutviklerne ikke hadde noen tro på tidsestimaterne i det originale prosjektestimatet, og at man så at ting måtte ta mye lenger tid enn det som var planlagt. Det viste seg også at prosjektet ikke i tilstrekkelig grad hadde fulgt opp kravspesifikasjonen eller dokumentert hvilke krav som var tilfredstilt og hvilke som gjensto. Etter å ha fått klarhet i kravspesifikasjon og brukerhistorier så gjorde man et nytt estimat og ny produksjonsdato ble så satt til 5. januar 2013, ett år etter den opprinnelige planen.

6.1.2 FDOK

FDOK er ett av om lag 30 prosjekter i Skatteetaten og det rapporteres til Skattedirektøren om fremdriften i prosjektet. FDOK ble gjennomført med en delt prosjektgruppe med ett team i Oslo og ett team i Grimstad.

FDOK skal utvikle felleskomponenten i den nye dokumentløsningen. FDOK skal tilby ett felles grensesnitt og innkanal mot Printmiljøet. Tjenesten skal samtidig tilby tjenester til fagsystemene slik at de kan koble sine utskriftsbehov til sentral print. FDOK er ment å være en felleskomponent som alle dokumenter og brev går gjennom med sporing av prosessen.

FDOK hadde i første fase en ekstern prosjektleder. Noen deltakere mente i ettertid at vedkommende hadde litt lite teknisk bakgrunn for et slikt systemutviklingsprosjekt og at prosjektlederen manglet den nødvendige forretningsforståelsen for hvordan Skatteetaten opererer. Utfordringer som blir nevnt i den sammenhengen er manglende kunnskap om hvordan man bestiller ressurser, hvilke avdelinger man må forholde seg til, hvilke ressurser man har tilgang til, hva som koster prosjektet penger i intern fakturering og hvilke lisenser man må betale for på testmiljøer og lignende. Dette førte til at prosjektet ikke satte av nok ressurser og ikke bestilte testmiljøer tidsnok. Prosjektet ble derfor forsinket mer enn nødvendig.

Det ble leid inn konsulenter i FDOK med kompetanse på systemutvikling i Java og på arkitektur. Dette var kompetanse som Skatteetaten hadde lite av fra før.

Det ble ikke kjørt noen kickoff i forbindelse med opprettelsen av FDOK. For miljøet i Grimstad opplevdes dette som rart. Da de ikke var involvert i den tidlige fasen av prosjektet så ble samarbeidsklimaet dårlig og man følte seg satt på sidelinjen av konsulentene. Det var miljøet i Grimstad som hadde utviklet en tidlig Proof of Concept-løsning. Det opplevdes som uheldig at prosjektet var delt i to team som satt på hver sin lokasjon i Oslo og Grimstad. «*Avstand har noe å si*» uttalte en systemutvikler i et av intervjuene. Man opplevde at konsulentene hadde mye kompetanse med samarbeidet lite. Det ble prøvd å gi beskjed om dette fra miljøet i Grimstad, men man fikk ikke gehør.

Over tid ble prosjektdeltakere i Grimstad faset over på andre prosjekter. Det er bare én systemutvikler i Grimstad som har vært med i prosjektet gjennom hele perioden.

Det var en del utskifting av konsulenter og arkitekter i prosjektet. Dette skapte utfordringer da de nye konsulentene måtte settes inn i prosjektet og opparbeide seg nødvendig forretningsforståelse før de kunne bidra.

Fra fagsystemene har man vært frustrert over manglende informasjonsflyt og involvering fra FDOK. Det har vært arrangert møter mellom FDOK og andre enheter som Arkitekturenheten og Sikkerhet uten at man har involvert systemutviklere fra fagsystemene. Dette har ført til at man blant annet har blitt enige om sikkerhetskrav som fagsystemet opplever at de ikke har mulighet til å tilfredsstille. Man måtte da kjøre nye møter og bli enige om hvordan det var mulig å skape et sikkert system.

Utfordringene i FDOK ved første produksjonssetting var manglende ytelse og manglende robusthet i løsningen. Det fantes heller ikke noen oversikt over hvilke krav som var tilfredsstilt og hvilke som gjensto i forhold til kravspesifikasjonen.

Deltakerne sier at de opplevde at FDOK ikke hadde kontroll på produktkø og prioritering av elementer. Man hadde manglende fokus på prosesser i perioden fra man var nesten ferdig med utvikling til man kunne gå i produksjon. FDOK fikk ikke tilgang til realistiske testmiljøer blant annet på grunn av manglende planlegging. Man opplevde at FDOK hadde et fokus på å levere koden fremfor å levere et komplett system. FDOK havnet på etterskudd og prioriterte å levere den funksjonaliteten som var lovet. Dette gikk på bekostning av å sikre tilgang til testmiljøer og produksjonsmiljø, og å levere en komplett fellestjeneste.

FDOK skulle overleveres til ny intern prosjektleder da tjenesten skulle gå i produksjon i slutten av 2011. En minimal versjon av FDOK ble satt i drift i november 2011, men med klare mangler. MDOK er i dag online mot FDOK og får dokumenter og maler, og skriver ut lokalt. Kravspesifikasjon sa at de også skulle få tilgang til sentral print av batcher tilsvarende deres behov, men dette er ikke implementert enda. Foreløpige mangler i FDOK innbefatter ytelseskravet på 800.000 dokumenter per døgn, robusthet i forhold til feilsituasjoner, driftbarhet og grensesnitt opp mot sporing av dokumenter.

Man har gjennomført en ny estimering av hva som gjenstår og hvordan man skal utvikle dette. Foreløpig plan tilsier at dette skal løses og settes i drift 5. januar 2013.

Etter at den nye prosjektlederen tok over kom man frem til at det ikke var hensiktsmessig at prosjektene utviklet parallelt. Den parallelle utviklingen førte til en del uheldige forhold, som at man fra fagsystemene testet funksjonalitet i felleskomponenten som man visste ikke var utviklet enda. Man valgte på bakgrunn av dette å la fagsystemene utvikle det som gjensto av intern funksjonalitet mens man ventet på at FDOK skulle utvikle det produktet som man hadde forutsatt i kravspesifikasjonen. På denne måten så får FDOK litt mer ro til å utvikle en robust tjeneste, mens MDOK og SDOK kan fokusere på sine egne komponenter.

Den virkelige verdien av FDOK vil man se når flere og flere benytter seg av løsningen. Det er i dag flere fagsystemer som venter på å koble seg til FDOK og den nye dokumentløsningen. Nye fagsystemer vil koble seg opp mot FDOK etter hvert som de har et funksjonelt behov for det.

6.1.3 MDOK

Behovet for en ny dokumentløsning ble synlig i MVA i 2009. Brukerne av MVA i de 5 regionene har i mange år uttrykt ønske om å få benytte Word i brevløsningen da den tidligere brevløsningen gjorde det vanskelig å redigere brev og hadde en del utfordringer relatert til spesialtegn i dokumentene. Å endre dette ville være en kjempejobb da MVA har over 500 dokumentmal der en hver endring må gjøres i hver enkelt dokumentmal.

Man fikk i oppdrag å finne en varig løsning på brevløsningen og høsten 2010 satte man ned et forprosjekt som så på hvilke muligheter man hadde.

Da prosjektet startet opp kjente man ikke til I-plattform, og man trodde at man skulle opprette en punkt-til-punkt forbindelse mellom MDOK og FDOK. Informasjonen om I-plattform uttaler man at kom inn over sidelinjen underveis i prosjektet. Dette førte til at man måtte gjøre endringer i prosjektet og man opplevde at det var begrenset informasjon og erfaringer om hvordan man skulle gjøre dette. Dette førte til at man støtte på en del utfordringer og problemstillinger som man ikke var klar over på forhånd. Man uttaler at *«det hadde vært en fordel om ting hadde fått litt bedre tid til å gå seg til slik at man fikk luket ut alle feil»*.

Selve prosjektet MDOK startet mot slutten av 2010. Planlagt produksjonsstart var i mars 2011, men komponenten kom først i produksjon i november samme år i en minimal versjon.

MDOK er en tjeneste i fagsystemet MVA som produserer dokumenter mot grensesnittet som FDOK tilbyr. Dokumentet lages ved at man kobler en valgt dokumentmal til et xml-uttrekk fra databasen som så blir til et dokument som sendes til FDOK for videre behandling.

MDOK utviklet ikke tjenesten som et dedikert prosjekt. MDOK ble kjørt over eget driftsbudsjett og som en del av linjen i MVA. Man hadde ikke et fastsatt budsjett eller et fast antall personer som jobbet med tjenesten. Det var fra 2 til 12 personer som jobbet på prosjektet avhengig av annen belastning i avdelingen. MVA har vært vant til å jobbe i en silo der man løser utfordringene som man har i produktkøen så snart man får mulighet. Da MDOK startet opp satte man en person som prosjektleder for MDOK som et internt prosjekt, men hun fikk ikke dedikerte ressurser til oppdraget. Dette har ført til at man underveis har mistet ressurser i perioder når det har vært oppdrag på driftssiden i systemet. Produksjon har hatt høyere prioritet enn MDOK i perioden.

MDOK led litt under at de ikke var organisert som et dedikert prosjekt med egne ressurser. Prosjektleder forteller at hun flere ganger mistet personell når det var utfordringer på driftssiden. Drift og produksjon gikk alltid foran når det var lite ressurser. Her har de i den senere tid blitt flinkere til å dedikere personell på prosjekter med en viss prosentandel. Dette har gjort at prosjektleder lettere kan planlegge og styre ressursene innenfor den prosentandelen de er tildelt.

Man benyttet ikke NUPAS eller Aurora i MDOK, men videreutviklet fagsystemet med samme arkitektur som lå i MVA fra tidligere.

MDOK og SDOK har lite med hverandre å gjøre på systemutviklernivå. Man utvikler mye parallell funksjonalitet, men samarbeid har bare forekommet sporadisk på private initiativ.

MDOK og SDOK har testet funksjonalitet i FDOK og «brøytet veien» for senere fagsystemer. Deltakerne fra MDOK uttaler at de opplever det som uheldig at prosjektene kjøres i parallell da man utvikler mot et system i stadig utvikling og bevegelse. Man uttaler at det ville vært lettere hvis FDOK var ferdig og tilbød definerte grensesnitt mot fagsystemet, i stedet for å være i utvikling og endring. Det opplevdes dog som en fordel at man kunne komme med innspill til FDOK underveis i forhold til hvordan tjenesten ble utviklet.

Systemutviklere i MDOK uttaler at *«det merkes at organisasjonen er i en læringsprosess med tanke på tjenesteorientert arkitektur»*. Systemer skal kobles mot hverandre og det finnes mye man enda ikke har tenkt på, og som man må finne løsninger på. Eksempelvis så vil transaksjoner settes i gang i fagsystemet mens brevene overføres til, og feilene skjer i, et annet system. Man må da spesifisere feilmeldinger og følge opp brevene mellom to systemer der man tidligere hadde hele flyten i en silo. Dette gir et nytt lag av kompleksitet, og man støter daglig på nye utfordringer på grunn av dette.

Deltakere fra prosjektet uttaler at prosjektet har blitt mye mer komplekst enn hva man trodde før oppstarten. Applikasjonen MVA er i seg selv veldig kompleks, og når dette skal kobles mot andre tjenester så fikk man utfordringer. Man oppdaget blant annet under utvikling at man mistet referanser til og kontroll på brevene sine mellom MDOK og FDOK, noe som kunne ført til at brukerne fikk flere eksemplarer av hvert brev. Skatteetaten er svært avhengig av tillit fra brukerne, og det stilles høye krav til datakvalitet og nøyaktighet. De fleste husker hvilket oppstyr man fikk rundt skatteoppgjørene i 2011 og 2012 da AltInn ga mange brukere tilgang til skatteoppgjøret til en tilfeldig person og løsningen gikk ned på grunn av for høy belastning (Stenerud & Thorvaldsen, 2012). Dette viser hvor høye krav som stilles til Skatteetatens sikkerhet og ytelse.

Den gamle brevløsningen eksisterer enda side om side med den nye brevløsningen mens man jobber med å flytte brevmalene over til den nye dokumentløsningen. Saksbehandler får opp ny eller gammel løsning avhengig av om malen er konvertert eller ikke.

Prosjektet MDOK har i dag 150-160 brevmaler i produksjon og cirka 140 brevmaler som er ferdige og klar for produksjonssetting. Løsningen har foreløpig vært noe ustabil slik at man nå har stoppet videreutvikling før man har fått løst en del stabiliseringstiltak. Man har foreløpig ikke mulighet for sentral print, men dokumentet sendes via FDOK for å gi statusinformasjon og sporing.

Man har utviklet nye brevmaler på bakgrunn av råd og retningslinjer fra firmaet Arkitektst. Firmaet er leid inn som en del av den nye språkprofilen til Skatteetaten, der dokumenter til kundene skal gjøres mer informative og mindre juridiske. Man søker også å få en mer enhetlig utforming av dokumentene i etaten. Det har oppstått en del motstand mot de nye brevmalene i organisasjonen. I MVA opplever man at saksbehandlerne ofte redigerer brevene ved å kopiere teksten fra tidligere brev slik at de får inn den informasjonen som de selv ønsker. Dette fører da til at man omgår den nye språkprofilen til Skatteetaten. MVA har derfor startet en aksjon der man skal gå ut med bedre informasjon om den nye språkprofilen og at det ikke er lov til å endre på dokumentene som er opprettet på bakgrunn av denne.

MVA har kjørt to løp i perioden mellom første produksjonssetting og andre produksjonssetting på nyåret 2013. Det kjøres et løp som de kaller forvaltningsløpet som har MDOK i produksjon mot den første versjonen av FDOK. Samtidig kjører de et systemutviklingsløp med testing mot den nye

versjonen av FDOK. Prosjektleder har da noen som jobber med forvaltning og noen som jobber med systemutvikling.

I MDOK hadde man ikke noen klare tidsplaner eller milepæler for prosjektet. Det ble uttalt at de hadde en tanke om at prosjektet skulle produksjonssettes i løpet av våren 2011.

MDOK bommet på tidsplanen, og prosjektleder sier at planleggingen av prosjektet ble for dårlig. Avdelingen har lært mye og gjort store endringer på hvordan man planlegger prosjekter i tiden etter at MDOK ble startet. Dette har vært en naturlig utvikling av avdelingen. Tidligere hadde de ikke noen klare strategier eller planer for hvordan MVA skulle utvikles. Man fikk inn behov fra brukerne og effektuerte disse så snart man fikk mulighet da man var vant til å fungere som en lukket silo. I den senere tid har MVA fått mer system på hvordan bestillingene kommer inn og hvordan de organiserer seg og planlegger systemutvikling og drift frem i tid.

6.1.4 SDOK

I 2009 begynte Sofie å se at de trengte en ny dokumentløsning. Det var blant annet høye lisenskostnader knyttet til den eksisterende løsningen, og man hadde lite mulighet til å endre utformingen av brev.

MDOK og SDOK har lite med hverandre å gjøre på systemutviklernivå. Man utvikler mye parallell funksjonalitet, men samarbeid har bare forekommet sporadisk på private initiativ.

SDOK brukte mye tid før prosjektet startet på å analysere hvilken effekt prosjektet ville ha på fagsystemet. De estimerte å bruke om lag 22.000 timer på prosjektet. SDOK hadde en detaljert prosjektplan med planlagt produksjonssetting i april 2012. Denne har de etter eget utsagn holdt, med unntak av en forsinkelse på en måned som følge av en forsinkelse i FDOK. Sofie består av mange tidligere konsulenter fra CapGemini, og mange har derfor lang erfaring med systemutviklingsprosjekter og planleggingsmetodikk.

Deltakerne uttaler at de har opplevd utfordringer i forhold til kommunikasjon mellom prosjektene underveis. Spesielt var det utfordrende at FDOK ikke tok med fagsystemene i viktige møter.

SDOK opplever ikke at lokasjoner var noe stort problem. Videoløsningene som man benytter fungerer greit, men de uttaler at det er den grunnleggende kommunikasjonen som kunne og burde vært bedre. Kommunikasjonen mot Print har ikke vært god nok, men her opplever SDOK at Print ikke har hatt tid til å prioritere dem. SDOK og MDOK har ikke hatt et godt nok samarbeid, og deltakerne mener at dette ville nok vært bedre hvis de var samlokalisert.

SDOK uttaler at de har benyttet samme prosjektmetodikk som man pleier å benytte i Sofie. De benytter Scrum som systemutviklingsmetode med produktkøer og korte sprinter, og føler de har god kontroll på dette.

6.1.5 SporDok

SporDok er et prosjekt som skal gi mulighet til å spore hvert enkelt dokument gjennom hele printprosessen. Dette gjøres ved at man legger ved en bestillingsfil til printfilen. Bestillingsfilen

inneholder detaljert informasjon om hvert enkelt dokument i filen. Denne informasjonen kan brukes i hvert ledd til å kontrollere at det som kommer ut av prosessen er det samme som ble bestilt. Man prøvde å finne en kommersiell leverandør som kunne gi denne funksjonaliteten. Da ingen av leverandørene kunne levere dette, valgte man å utvikle SporDok selv. SporDok tilbys også mot eksterne kunder.

SporDok krever at fagsystemene leverer en bestillingsfil med oversikt over hva som ligger i printfilen. Dette gir SporDok anledning til å kunne spore utskriften gjennom alle sine leveransepunkter slik at man vet den nøyaktige flyten til hver utskrift.

SporDok ser det ikke som hensiktsmessig å følge hvert enkelt dokument og oppdatere på dette med tanke på hvor mange kall dette vil innebære per printjobb. De mener det vil være mest hensiktsmessig å gjøre denne sporingen på hver enkelt batch.

SporDok må utvikle deler av funksjonaliteten som var tiltenkt å skulle ligge i FDOK. De må blant annet selv finne en løsning for å kunne legge inn et tilgjengelig fontbibliotek til bruk i utskriftene.

SporDok føler at Dokumentprosjektet ikke i tilstrekkelig grad respekterte kompleksiteten i Print. SporDok følte at det var en holdning som tilsa at hvis man løste FDOK så skulle man alltid klare å få til utskriftene. Print-miljøet poengterer at det er en forskjell i kompleksitet mellom å skrive ut ett dokument på en lokal printer og utskrifter i 100.000-arksklassen med konvoluttering og pålitelighetskrav.

SporDok uttaler også at teknologivalget i FDOK er mislykket da de til dags dato ikke klarer å levere printfiler av riktig størrelse. SporDok har et veldig sentralt krav om å få filer som er tilpasset størrelsen på deres arkruller for å kunne optimalisere ytelsen, men dette er ikke løst enda.

6.2 Forventede effekter av Dokumentprosjektet

Man venter å få flere effekter ut av Dokumentprosjektet. Dokumentprosjektet skal først produksjonssettes 5. januar 2013 etter den nye planen, men man har alt gjort seg noen erfaringer med mellomløsningen. En av deltakerne uttalte at «*Den store verdien av en felleskomponent oppnår man i det alle systemene bruker den, og da får man den virkelig store effekten*». Dokumentprosjektet har gått over både budsjett og tidsrammer som opprinnelig var lagt, men det er viktig å huske på at de grunnleggende kostnadene ved å gå over til SOA ofte vil være høye, mens man får effektene over tid (BEA, 2005).

Her presenteres noen av de forventede effektene Skatteetaten håper å få fra Dokumentprosjektet.

6.2.1 Skatteetaten

Skatteetaten som helhet vil oppleve at man kjører flere utskrifter via sentral print. Dette vil føre til mindre arbeid for saksbehandlerne i fagsystemene og gi en felles prosess for dokumenter. Man vil også få en bedre oversikt over alle de brevene som eksisterer i de ulike fagsystemene da disse nå samles og får en fornyet visuell og språklig profil.

Det vil ligge mer funksjonalitet i dokumentløsningene som sporing og arkivering. Dette kan potensielt gi besparelser og bedre informasjon. Til sist så vil man oppnå bedre sikkerhet og pålitelighet ved at

man nå får en gjennomgående tjeneste der sikkerheten er ivaretatt og man automatiserer oppgavene med sporing i hvert ledd

6.2.2 Skatteetatens Service- og IT-partner (SITS)

Driftsavdelingen (SITS) vil se at man kan få lavere vedlikeholdsutgifter ved at FDOK driftes som en sentralisert tjeneste. Dette vil redusere behovet for dokumentløsninger i fagsystemene. Ved å rydde opp i silo-strukturen og «spagetti-integrasjonene» der vil man få en enklere sentralisert tjeneste. Dette vil igjen gi en bedret forvaltning av systemene og potensielt en mer helhetlig endringshåndtering.

6.2.3 Print-miljøet

Print-miljøet har tidligere måttet forholde seg til mange kunder i Skatteetaten og hos samarbeidspartnere der man fikk inn utskriftsjobber i mange ulike formater. Etter Dokumentprosjektet vil man kunne forholde seg til kun én innkanal for utskrifter. Dette gir dem en mulighet til å kunne standardisere inputene.

Print vil kunne tilby sporbarhet av alle dokumenter helt frem til postleggelse og mulighet for returskanning ved hjelp av FDOK og SporDok. Dette vil gi en mye bedre integritet der man tidligere var avhengig av en del manuelle operasjoner for å sikre at dokumentene ble sendt korrekt.

6.2.4 Fagsystemene

Fagsystemene vil oppleve bedret funksjonalitet for saksbehandlerne gjennom den nye dokumentløsningen. De får tilgang til dokumentene i en fornyet visuell og språklig profil og de får også muligheten til å kunne editere brevene før de sendes ut til mottakerne.

Man får muligheten til å benytte sentralisert utskrift i den nye løsningen med sporing av dokumentet. Tidligere måtte saksbehandler i noen av fagsystemene skrive ut dokumentet, legge det i konvolutt, adressere og sende det til mottaker. Med den nye løsningen kan dette gjøres sentralt. Så vidt deltakerne kjente til så foreligger det ingen planer eller tanker for hvordan man tenker å utnytte denne tidsbesparelsen.

6.2.5 FDOK

Med tiden vil man i FDOK implementere mulighet til å kunne velge alternative utkanaler. Man har planer om å kunne benytte elektroniske utkanaler som sms, epost og AltInn med tiden. Utvikling av denne funksjonaliteten var tidligere tenkt å løses i Dokumentprosjektet, men er i dag flyttet ut i et prosjekt som heter PIPPI.

6.2.6 Skatteyttere

Skatteytterne vil få dokumenter fra Skatteetaten med bedre layout og bedre informasjon om hvordan de kan kontakte skatteoppkrever i etterkant. Informasjonen vil bli mer forståelig for menigmann enn hva den var tidligere på grunn av den nye visuelle og språklige profilen.

6.3 utfordringer underveis i prosjektene

Selv om Dokumentprosjektet lyktes med å levere en felles dokumentløsning har det oppstått en del utfordringer underveis i prosjektet. Det er viktig å belyse dem da de har hatt innvirkning på fremdriften og løsningene som er blitt valgt. I et sosio-teknisk perspektiv kan disse utfordringene forklares ved å se på samspeillet mellom menneskene og teknologien som skal utvikles. I fremtidige prosjekter vil man da kunne hindre at tilsvarende utfordringer oppstår ved tidlig å fokusere på å motvirke årsakene.

6.3.1 Sikkerhet

Skatteetaten er en etat som er sterkt avhengig av publikums tillit. Det stilles derfor høye krav til sikkerheten i løsningene deres. I Dokumentprosjektet fikk man meget høye sikkerhetskrav fra Sikkerhetsavdelingen. Med tanke på at trafikken går internt på lukkede systemer så synes flere av systemutviklerne at disse sikkerhetskravene var i overkant høye. Kravene har ført til at systemutviklerne har måttet utvikle tjenester med høyere sikkerhetskrav og kompleksitet enn hva de mente var nødvendig.

6.3.2 Ytelse

Ytelsen har vært en utfordring i Dokumentprosjektet. Kravet sier at man skal kunne produsere cirka 800.000 dokumenter i løpet av én natt. Det jobbes med å optimalisere prosessen for å dekke dette kravet.

I-plattform har vist seg å være en flaskehals for transport av filer. Når man overfører dokumenter over I-plattform har man merket at det tar forholdsvis lang tid å overføre filene. Etter mye testing har man kommet frem til at ytelsen blir best hvis man overfører rundt 50 dokumenter i hver pakke. Det blir veldig mange pakker hvis det er snakk om brev som skal ut til alle landets skatteyttere.

Kravene fra Print-miljøet har vært filer med ca. 60.000 dokumenter for å optimalisere benyttelsen av ark-ruller, men dette har man foreløpig ikke klart å løse i FDOK.

6.3.3 Kompetanse

SOA-programmet i Skatteetaten har pågått siden 2008 (Staverløkk, 2009), men Dokumentprosjektet omtales som den første virkelige felleskomponenten som utvikles i organisasjonen. Det er utviklet en integrasjonsplattform (I-plattform) og en systemutviklingsplattform (Aurora). Det er fortsatt begrensede erfaringer med bruk av disse komponentene, og en systemutvikler sier at: *«Det merkes at organisasjonen er i en læringsprosess med tanke på tjenesteorientert arkitektur. Systemer skal kobles*

opp mot hverandre, og det finnes mye man enda ikke har tenkt på i organisasjonen og som man må finne løsninger på».

Skatteetaten innførte den nye systemutviklingsplattformen basert på Java like før Dokumentprosjektet startet opp. Skatteetatens systemutviklere har tidligere utviklet fagsystemene innenfor hver sin silo og brukt veldig mange ulike programmeringsspråk og arkitekturer. Aurora utviklet retningslinjer for hvordan plattformen skulle brukes, men dette skjedde parallelt med, og delvis i etterkant av, Dokumentprosjektets utvikling. I dag må alle som skal utvikle på Auroraplattformen gjennomgå et utdannings- og sertifiseringsprogram, men dette eksisterte ikke i de tidlige fasene av Dokumentprosjektet. Dette førte til en del utfordringer der man av og til måtte utvikle ting på nytt fordi Aurora kom med nye føringer.

Da systemutviklerne skulle utvikle Dokumentprosjektet i Java var dette et språk som de færreste hadde benyttet tidligere. Dette ble en utfordring for mange, og flere deltakere uttaler at de tror Skatteetaten har undervurdert utfordringen dette kompetanseløftet innebar.

Skatteetaten leide inn konsulenter i flere sentrale roller i prosjektet. Konsulentene hadde bred erfaring med Java og ble leid inn som prosjektleder, arkitekter og systemutviklere. Dette ga prosjektet en økt kompetanse på bruk av Java, men deltakerne uttalte likevel at man har hatt utfordringer med systemutviklingsspråk og systemutviklingsplattformen i Dokumentprosjektet.

Konsulentene tilførte prosjektet viktig kompetanse innen Java og systemutvikling, men de fikk en utfordring i forhold til å ha den riktige forretningsforståelsen av Skatteetaten. Skatteetaten er en kompleks og byråkratisk organisasjon. Forståelse av organisasjonen og hvordan denne fungerer synes å være veldig viktig for å forstå hvilke enheter man skal forholde seg til og hvilke ressurser man har tilgang til. Konsulentene manglet denne kompetansen. Dette gjelder spesielt i FDOK der prosjektlederen selv var ekstern konsulent. Den manglende forretningsforståelsen førte til forsinkelser for prosjektet ved at man ikke fikk bestilt korrekte testmiljøer til FDOK. Det er for eksempel slik at man må gjennomføre flere tester på forskjellige nivåer, blant annet en test på reelle testdata, før man får lov til å introdusere ny kode i et fagsystem. For å få tilgang til disse miljøene må dette bestilles, og det koster penger for prosjektet. I FDOK kjente man ikke godt nok til dette, og man klarte ikke å få tilgang til de riktige testmiljøene til rett tid. Dette førte til forsinkelser i prosjektet.

Utover dette så er det mange avdelinger å forholde seg til ved utvikling av en kompleks komponent som griper inn i mange deler av Skatteetaten. Rollen som prosjektleder i FDOK beskrives av flere deltakere som en vanskelig rolle å fylle som eksternt innleid konsulent. Dette fordi den krever mye kompetanse og forretningsforståelse av organisasjonen.

Manglende forretningsforståelse førte også til utfordringer i form av at man hadde møter med avdelinger i Skatteetaten på vegne av alle prosjektene der man avtalte føringer som delprosjektene ikke kunne innfri. Systemutviklerne i FDOK som satt i Grimstad ga tidlig i prosjektet beskjed om at de mente at man hadde for lite forretningsforståelse i prosjektledelsen, men de opplevde ikke å få gehør for dette argumentet.

De gangene man byttet ut konsulenter i noen av rollene måtte den nye konsulenten opparbeide seg forretningsforståelse i sin rolle. Dette førte til at man fikk produktiviteten ble lavere i en periode hver gang en konsulent ble byttet ut.

Skatteetaten uttaler at de har som mål å skulle ha kompetanse internt til å kunne gjennomføre de daglige oppdragene, og leie inn ressurser i perioder med spesielt stor belastning. De ønsker å ha nødvendig kompetanse til å kunne dekke de viktigste rollene selv. I Dokumentprosjektet ble de nødt til å leie kompetanse til å fylle flere roller.

Dokumentprosjektet er et forholdsvis stort og omfattende prosjekt som skal utvikle en brevløsning som kan tas i bruk av alle Skatteetatens systemer. I dette prosjektet startet man opp i en ganske vanskelig fase der Skatteetaten nettopp har gjennomført relativt store endringer. En ny integrasjonsplattform og en ny systemutviklingsplattform som man enda ikke hadde mye erfaringer med ville skapt en del utfordringer for mange organisasjoner. Spesielt siden den nye systemutviklingsplattformen innebærer at man skal benytte et nytt systemutviklingspråk.

Skatteetaten har økonomi og mulighet til å kunne utvide budsjett og tidsramme slik at prosjektet likevel gjennomføres og gir den ønskede effekten.

6.3.4 Koordinering mellom prosjektene

Dokumentprosjektet startet opprinnelig opp som fire uavhengige prosjekter som samarbeidet gjennom FDOK og kravspesifikasjonene der. Interessenter fra alle prosjektene samt viktige sentrale enheter i Skatteetaten møttes jevnlig for å koordinere innsatsen. Disse møtene ble i følge deltakerne holdt ved behov.

Felleskomponenten FDOK oppfattes som navet i Dokumentprosjektet i den tidlige fasen. Det er her man lager kravspesifikasjonen og det er prosjektleder for FDOK som styrer koordineringsmøtene. Det at prosjektleder i FDOK var innleid hadde sine egne utfordringer. Vedkommende skulle koordinere innsatsen i prosjektene, men manglet forretningsforståelse fra Skatteetaten. Prosjektlederen manglet derfor litt av forståelsen for hvordan avgjørelser tas i etaten.

I fasen etter at kravspesifikasjonen var klar, og man var enige om grensesnittene, ble møtene holdt sjeldnere. Deltakerne uttaler at man stolte på kravspesifikasjonen og at prosjektene skulle levere den funksjonaliteten de var blitt enige om. Det ble holdt noen koordineringsmøter, men det synes som om det ble for få av dem, og at man ikke fikk delt den nødvendige informasjonen på riktig tidspunkt.

En deltaker uttalte: *«Et tidlig fokus på grensesnitt og en enighet om hva det er og hvordan det virker er viktig»*. Årsaken til denne uttalelsen var at i FDOK endret dette seg mer enn ønskelig. Dette hindret i neste omgang fremdriften hos brukertjenestene.

FDOK holdt møter med sentrale avdelinger i Skatteetaten uten å ta med deltakere fra fagsystemene. Dette førte til at man avtalte krav som fagsystemene ikke kunne overholde, og uten at man informerte dem om de nye kravene. Endringer fra FDOK var ikke godt nok kommunisert og fagsystemene måtte gjøre om arbeid når de fikk informasjonen. *«Vi har sett at det er viktig at felleskomponenten lytter til innspill fra involverte parter og eksterne enheter og kommuniserer dette ut til alle deltakerne»* sa en deltaker.

Det synes som at det har vært for lite fokus på koordineringen i Dokumentprosjektet. Prosjektet besto av fire delprosjekter som utviklet sine tjenester parallelt. Det ble holdt noen

koordineringsmøter, men veldig mye av systemutviklingen ble gjort på bakgrunn av den skriftlige kravspesifikasjonen.

Prosjektene benyttet Scrum som systemutviklingsmetode og Cohn (2010) argumenterer for at man i smidige metoder legger til rette for en tettere kommunikasjon rundt systemutviklingen. Han poengterer at skriftlige kravspesifikasjoner altfor lett mistolkes og det er uklarheter rundt hva behovet egentlig er. Enkelte av deltakerne var inne på at man hadde uklarheter til for eksempel om ett dokument var det samme som ett ark eller som ett brev. Etter at man gikk over til å bruke brukerhistorier i større grad, så opplevde man at man fikk en bedre forståelse for hva som egentlig lå bak kravene.

Koordineringen som ble gjennomført mellom prosjektene var hovedsakelig på prosjektledernes nivå. Man hadde noen møter der man oppdaterte hverandre på fremdriften i prosjektene. En av prosjektlederne uttalte at det var mange saker underveis som falt mellom to stoler og at man ikke helt visste hvem som skulle ta tak i dem. Uten et klart mandat var det uklart hvem som kunne ta avgjørelser i disse sakene. Prosjektlederen sa at det ville vært naturlig at prosjektleder i FDOK gjorde dette, men at han hadde det såpass travelt med å styre sitt eget prosjekt at ting gjerne ble liggende.

Prosjektlederne opplevde at de ble presset på tidsfristene fra ledelsen. Omorganiseringen av SITS gjorde at det til tider var utfordrende å finne ut hvem som hadde ansvar for enkelte områder i Skatteetaten. Tidspresset på leveransen av FDOK høsten 2011 opplevdes belastende for noen av prosjektlederne. De fikk stadige spørsmål om status og opplevde et press på å levere et produkt selv om funksjonaliteten ikke var god nok. En deltaker uttalte at man ikke ville satt systemene i produksjon hvis det var opp til prosjektlederne selv å avgjøre dette.

Enkelte deltakere har fortalt at man organiserte litt feil da FDOK skulle ha fullt ansvar for felleskomponenten og design av denne, mens Sofie og MVA utviklet hvert sitt prosjekt. Prosjektoppbyggingen ga ingen formell overbygging over prosjektene, og prosjektene måtte koordinere arbeidet sitt selv. Det manglet en felles styringsgruppe som kunne ta avgjørelser og kontrollere fremdrift og valg. Dette tok man tak i da overbyggingen Dokumentprosjektet ble opprettet i desember 2011 med en egen styringsgruppe som skulle se på delsystemene som en helhet. Etter dette har man fått en bedre styring med prosjektene og koordinering mellom dem.

Dokumentprosjektet fikk en egen prosjektleder og egen styringsgruppe. Prosjektleders oppdrag var å koordinere aktivitetene mellom delprosjektene og sørge for at informasjonsflyten kom i gang. Etter at Dokumentprosjektet ble opprettet, rapporterer deltakerne at informasjonsflyten har vært mye bedre. Man har reorganisert FDOK og endret slik at man kjører mer sekvensielle løp i utviklingen. I tillegg har man ukentlige møter mellom prosjektlederne og deltakere fra Print og Drift. I disse møtene tar man opp hvilke aktiviteter som utføres i de ulike prosjektene og eventuelle avhengigheter til andre prosjekter, ganske likt «Scrum of Scrums» (Cohn, 2010). Deltakerne uttaler at de synes at koordineringen har blitt mye bedre etter at man innførte den felles overbyggingen og styringsgruppen.

6.3.5 Kommunikasjon

Kommunikasjonen har i følge deltakerne vært en utfordring i enkelte faser av Dokumentprosjektet.

Innad i FDOK kan det virke som om kommunikasjon og tillit kan ha vært litt fraværende i enkelte perioder. Miljøet i Grimstad hadde identifisert behovet for en dokumentløsning, og det var de som hadde utviklet POCen. Da FDOK ble startet opp ble den ansvarlige for POC-prosjektet foreslått som prosjektleder i FDOK. Dette var ikke ønsket av de andre miljøene, og man ansatte i stedet en ekstern prosjektleder. Han skulle lede et prosjekt som hadde systemutviklere både i Grimstad og i Oslo.

Deltakerne fortalte at det ikke ble holdt noen oppstartssamling eller kickoff der man ble kjent innad i FDOK. I stedet ble miljøet i Grimstad satt på sidelinjen en periode mens konsulentene restrukturerte prosjektet. Miljøet i Grimstad ble etter hvert fasett inn i prosjektet igjen, men inntrykket fra intervjuene er at samarbeidsklimaet i FDOK ikke var det beste. Miljøet i Grimstad meldte fra om at mente det var for lite forretningsforståelse i Oslo - uten å få gehør. Dette førte til en del frustrasjon. I løpet av FDOK var det bare én systemutvikler fra Grimstad som deltok i prosjektet gjennom prosjektets levetid. De andre systemutviklerne var litt «til og fra» avhengig av annen belastning i Grimstad.

Skatteetatens systemutviklingsmiljø er vant til å utvikle systemet innad i sitt eget fagsystem og sin egen silo. Etaten har en byråkratisk oppbygging og IT-arkitekturen kan på mange måter karakteriseres på samme måte og med de samme utfordringer som Baskerville et al. (2010) beskriver for bankene de studerte. Historikken har gitt Skatteetaten mange heterogene systemer basert på helt ulike arkitekturer og systemutviklingspråk som må samhandle for å dekke kundenes samlede informasjonsbehov. Mens de tidligere kun hadde sitt eget fagsystem og sine egne systemutviklingsrutiner å forholde seg til, så må man nå fokusere mer på kommunikasjon mellom team innad i ett systemutviklingsprosjekt. Siden Skatteetaten innfører SOA med felles integrasjonsplattform, felles systemutviklingsplattform og fellestjenester, så blir man også i større grad nødt til å fokusere på kommunikasjon og samarbeid mot andre interessenter. I Dokumentprosjektet har det kommet frem at man fortsatt har en del å hente på samarbeid og kommunikasjon mellom de ulike aktørene, spesielt på systemutviklingsnivået.

Enkelte av deltakerne uttaler at samarbeidsklimaet mellom MDOK og SDOK er blitt kritisert underveis i prosjektet. Systemutviklingen i de ulike prosjektene har foregått uavhengig av hverandre. Kommunikasjonen som har foregått mellom de to prosjektene synes stort sett å ha gått mellom prosjektlederne. Det har vært noe kommunikasjon på lavere nivåer om detaljer, men det har ikke vært mye direkte samarbeid. En av systemutviklerne uttalte at han visste at det satt en person i det andre prosjektet som utviklet den samme funksjonaliteten som han utviklet, uten at de har samarbeidet. Sannsynligheten er høy for at de to systemutviklerne vil støte på mange av de samme utfordringene og kan lære av hverandre så lenge begge jobber mot samme felleskomponent. Mye av idéen bak SOA er at man skal utvikle tjenester og byggeblokker som kan gjenbrukes (BEA, 2005). Likevel har man ikke i MDOK og SDOK gjort noen forsøk på å samarbeide på dette nivået for å kunne utvikle gjenbrukbare komponenter. Systemutviklerne som ble intervjuet mente at et slikt initiativ måtte komme som et pålegg fra prosjektledelsen hvis man skulle få til samarbeid i praksis. En av systemutviklerne hadde vært med på et slikt samarbeid på tvers av prosjekter i et tidligere prosjekt. Dette hadde fungert bra, og han var sikker på at det ville være mulig å få besparelser ved å gjøre det samme i Dokumentprosjektet.

Dokumentprosjektets organisering ligger ganske tett opp mot det som Cockburn (2007) beskriver som en multisite organisering. Man har relativt komplette systemutviklingsmiljøer på hver enkelt

lokasjon som utvikler sub-systemer i både MDOK, SDOK og SporDok. Unntaket her er at deler av FDOK hadde ganske få systemutviklere i Grimstad, mens hovedstyrken satt i Oslo. Denne oppbyggingen er veldig vanlig og fungerer godt så lenge prosjektlederne, eller ansvarlige i hver lokasjon, kommuniserer godt.

Flere av deltakerne uttaler at «*lokasjon har mye å si*» når de beskriver samarbeidsklimaet i prosjektet. De uttaler også at kommunikasjonen i prosjektet har vært utfordrende, spesielt i prosjektets tidlige fase. De har opplevd i prosjektene at de ikke har hatt den kommunikasjonen og det samarbeidsklimaet som de kanskje hadde ønsket. En av prosjektlederne sa at det var en stor forskjell for systemutviklerne når man ikke kunne gå inn på kontoret og snakke om løsningene når man hadde behov for det. Man benyttet telefon og videokonferanse, men det kan synes som at disse verktøyene ble benyttet mest til faste, planlagte møter. Den daglige kommunikasjonen og koordinering synes å ha vært litt mangelfull.

I Dokumentprosjektets tidlige fase hadde man mye fokus på en skreven kravspesifikasjon og kontrakter mellom prosjektene. Koordineringsmøtene ble holdt sjelden og deltakerne sier at de stolte på kravene. Det viste seg etter hvert at dette ikke var godt nok. Kravene ble gjerne tolket forskjellig og enkelte deltakere mente at man ikke fulgte kravene nøye nok. De sier at hvis man i stedet hadde møttes oftere, ville man hatt mulighet til å snakke om utfordringer og løsninger på en bedre måte og oppnådd en felles forståelse for hvordan komponentene skulle utvikles.

I den tidlige fasen av prosjektet når man utviklet kravspesifikasjonen hadde man flere møter med ulike interessenter. Det viste seg her at man manglet et felles vokabular og at man ofte snakket forbi hverandre om de samme utfordringene. Gjennom prosjektet fikk man etablert et felles vokabular som gjorde at man kunne omtale tjenestene og utviklingen på samme måte. Dette var en effekt Ordanini & Pasini (2008) også oppdaget i sin studie av organisasjoner som innførte SOA.

Omorganiseringen av SITS førte til at man på noen områder mistet kontroll på hvem som hadde ansvar for spesifikke fagområder i Skatteetaten. De man var vant med å kontakte hadde nå fått nye ansvarsområder. En av deltakerne uttalte at «*denne omorganiseringen har nok påvirket prosjektet mye mer enn ledelsen er klar over*». Ressurspersonene som man var vant til å benytte i organisasjonen hadde nå fått nye ansvarsområder, slik at man måtte lete seg frem til den nye ansvarlige.

6.3.6 Parallele prosjekter

Skatteetaten har i løpet av relativt kort tid gjennomført flere store prosjekter i SOA-programmet sitt (se punkt 4.3). Integrasjonsplattformen og systemutviklingsplattformen var begge relativt nylig innført da prosjektet startet, og man manglet retningslinjer og erfaringer for bruk av disse byggeblokkene. En av prosjektlederne uttalte at hun ikke kjente til integrasjonsplattformen da prosjektet startet opp. Prosjektdeltakerne trodde at de skulle benytte direkteintegrasjon mot felleskomponenten, og fikk en del ekstraarbeid da de ble klar over at de skulle benytte integrasjonsplattformen.

En av deltakerne uttalte om I-plattform at systemer ofte må få kjøre og fungere en stund før man får luket ut alle barnesykdommene. Det vil ofte være en del utfordringer med nye komponenter, og

kanskje spesielt med fellestjenester som skal brukes på tvers av organisasjonen. Det er indikasjoner på at de møter den samme utfordringen i forhold til bruk av systemutviklingsplattformen.

Systemutviklingsplattformen Aurora var fortsatt ikke ferdigutviklet da Dokumentprosjektet startet opp. Det ble besluttet at Dokumentprosjektet skulle benytte systemutviklingsplattformen og Java som systemutviklingsspråk. Når Dokumentprosjektet ble kjørt parallelt med at deltakere fra Aurora-prosjektet laget retningslinjer for bruk av plattformen, så skapte dette en del utfordringer. Retningslinjene og kravene kom i flere tilfeller etter at funksjonalitet var utviklet i Dokumentprosjektet. Dokumentprosjektet måtte da endre arbeid de allerede hadde utført for å tilpasse seg retningslinjene.

SITS ble som nevnt tidligere omorganisert høsten 2011. Dette førte til at prosjektene måtte bruke tid på å finne frem til hvem som hadde overtatt ansvaret for viktige ansvarsområder.

I tillegg til de sentrale endringene som ble gjort like før Dokumentprosjektet startet opp så ble FDOK, SDOK, MDOK og SporDok først startet opp som fire prosjekter med egne prosjektledere uten noen formell overbygging. Prosjektene pågikk parallelt slik at prosjektene utviklet sin tjeneste mot et grensesnitt som også var i utvikling. Flere av deltakerne nevnte dette som en kompliserende faktor. Den ene prosjektlederen forklarte utfordringen slik: *«Hadde FDOK vært ferdigutviklet og tilbudt oss sine tjenester så hadde vi bare hatt utvikling på ett sted. Men når det foregår utvikling på så mange steder samtidig så er det en stor utfordring»*.

6.3.7 Bruk av konsulenter i viktige roller

Skatteetaten uttaler at de vil ha tilstrekkelig med egne systemutviklere og arkitekter til å kunne håndtere daglig drift. I Dokumentprosjektet lyktes man ikke fullt ut med å få egne ansatte i flere av disse nøkkelrollene. Det finnes flere årsaker til dette, men den viktigste er kanskje at man i Skatteetaten ikke har så mye erfaring på systemutvikling i Java. Siden systemutviklingsplattformen er basert på Java måtte man leie inn konsulenter med denne kompetansen.

Miljøet i Grimstad ønsket som tidligere nevnt at man skulle ansatte prosjektlederen fra «Proof of Concept»-løsningen som prosjektleder for FDOK, men miljøet i Oslo ønsket ikke dette. Man gikk i stedet for å leie inn en ekstern prosjektleder i FDOK.

Flere ganger i løpet av prosjektet ble det byttet innleide konsulenter i FDOK. De nye konsulentene måtte settes inn i prosjektet før de kunne delta. Dette fikk uheldige konsekvenser siden man fikk flere stopp i fremdriften.

Den eksterne prosjektlederen i FDOK hadde ikke noen erfaring fra Skatteetaten. Skatteetaten er en kompleks etat med mange avdelinger som skal komme med innspill til nye systemer. Prosjektleder i FDOK har behov for å forstå hvordan Skatteetaten fungerer og hvem som leverer premisser for løsningene. I tillegg er det slik at man internfakturerer for tjenester, og man må da vite hvilke miljøer man må bestille fra og hvilke tjenester som vil koste penger for prosjektet.

Uten at man har belegg for å vite dette sikkert, så uttalte flere av deltakerne i studien at de tror løsningene i FDOK ville blitt annerledes hvis prosjektlederen hadde hatt en personlig tilknytning til Skatteetaten. De uttalte at man nok ville gjort andre valg hvis man visste at man skulle leve med

løsningene i 10-20 år. Andre mente at en ekstern prosjektleder lett styres av leveransen og kravene til denne, mens de mente at fokuset heller burde legges på hvilket behov Skatteetaten har.

Det ble byttet prosjektleder i FDOK på slutten av 2011. Dette var et planlagt bytte. Forutsetningene var at den nye prosjektlederen skulle overta forvaltning av en kjørende felleskomponent, men dette ble ikke tilfellet. Etter overtakelsen gikk den nye prosjektlederen gjennom kravspesifikasjoner og estimerer på nytt. Det ble besluttet at prosjektet måtte utvides til januar 2013 for å utvikle manglende funksjonalitet. Det viste seg at estimatene fra august 2011 ikke var korrekte.

Dokumentprosjektet ble innført som en overbygging over FDOK, MDOK og SDOK omtrent samtidig som man byttet prosjektleder i FDOK. Dette er også en ekstern konsulent, men vedkommende har vært innleid i Skatteetaten i mange år og kjenner organisasjonen. Denne rollen er en mer koordinerende rolle, og tilbakemeldingene fra deltakerne tilsier at han fungerer godt.

6.3.8 Prosjekt- og systemutviklingsmetode

Samtlige deltakere ble spurt om de brukte noen spesiell prosjektmetodikk ved et tjenesteorientert prosjekt, og om de brukte noen spesiell systemutviklingsmetodikk. Svarene de ga var at man benyttet en egentilpasset versjon av Scrum til systemutvikling. Man benyttet ikke noen spesiell prosjektmetodikk eller tok noen spesielle forhåndsregler ved ledelse og gjennomføring av et tjenesteorientert prosjekt.

Prosjektene benyttet Scrum som metode, men det synes som om det var varierende hvor mye av Scrummetoden man hadde tatt inn i hvert av prosjektene. Noen av deltakerne beskrev flere elementer av Scrum som man hadde implementert, mens andre uttalte at man nylig hadde begynt med Scrum og at man foreløpig hadde en minimal tilnærming.

Alle prosjektteamene sa at de hadde en Scrum master, og at de gjennomførte Scrummøter innad i prosjektet. Mellom prosjektene gjennomførte man koordineringsmøter ved behov, men disse ble det færre av etter at kravspesifikasjonen var ferdigstilt og man begynte å utvikle tjenestene.

Dokumentprosjektet er som nevnt tidligere et ganske stort prosjekt med mange involverte interessenter. I tillegg til dagens interessenter, så sitter det mange ansvarlige for andre fagsystemer og eksterne kunder på gjerdet. Disse planlegger å ta i bruk FDOK og dokumentløsningen så snart den er ferdigutviklet.

Informasjonsflyten i dokumentløsningen fremstår som en komplisert prosess. Det er mange interessenter inne i bildet og informasjonen skal flyte fra fagsystemene, via I-plattform, til felleskomponenten. Deretter skal informasjonen sendes videre til sentral utskrift med sporing av hvert steg i prosessen helt frem til utsending av post. I tillegg skal eventuelle feilsituasjoner rapporteres tilbake til saksbehandler eller driftspersonell.

Utover kompleksiteten i informasjonsflyten så skal også arkitekturen og systemutviklingen gjøres på en riktig og sikker måte. Dette stiller krav til prosjektene og måten disse gjennomføres på.

Etter at Dokumentprosjektet ble opprettet som en paraply over delprosjektene har man tatt tak i kommunikasjon og koordinering. Man har innført felles møter mellom prosjektene, ganske likt tankegangen fra «Scrum of Scrums» som Cohn (2010) anbefalte når man hadde flere team.

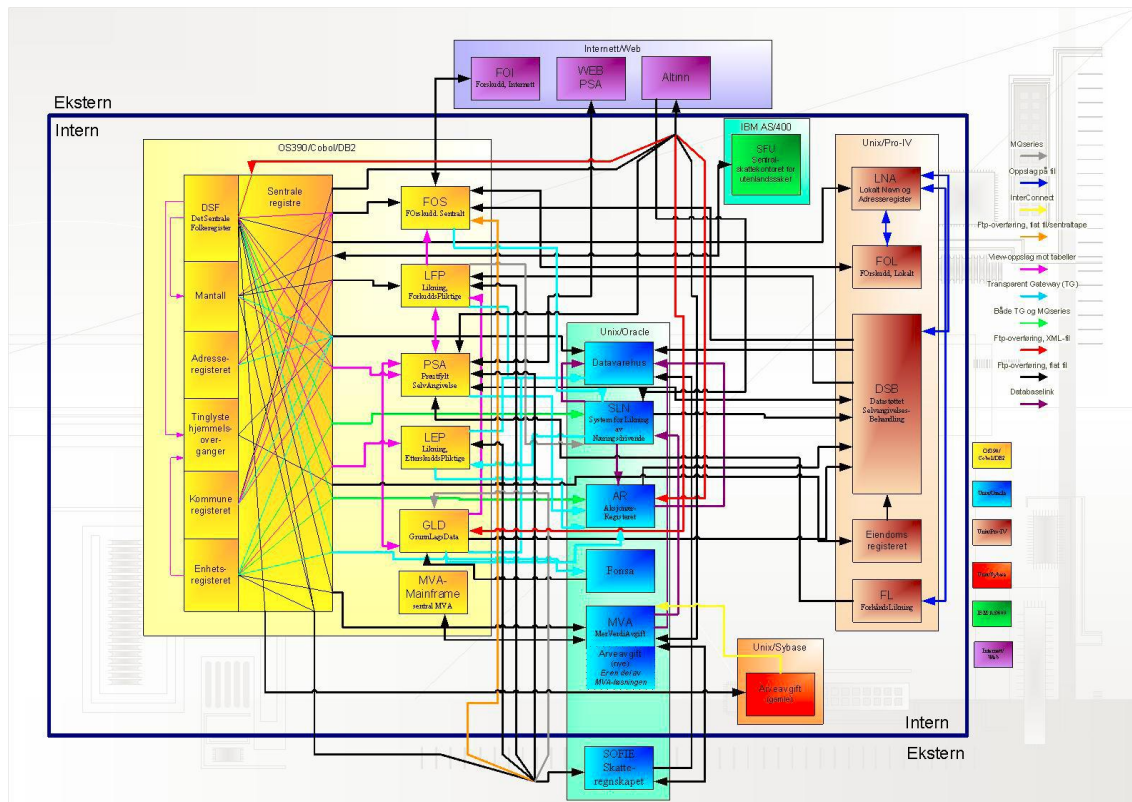
6.3.9 Komplexitet

Man har i Skatteetaten fått et økt kompleksitetsnivå i arkitekturen som følge av tjenesteorientering og nye utfordringer dette bringer med seg. Kompleksitetsnivå i Skatteetatens IT-arkitektur økte med en fersk satsing på SOA samtidig som silo-applikasjonene fortsatt er i bruk. Deltakere i studien uttalte at løsningene enda ikke var kvitt «barnesykdommene» og at man ikke hadde fått laget klare retningslinjer for systemutvikling og bruk enda.

I dokumentløsningen skulle flere systemer snakke sammen og dele informasjonsflyt og statusinformasjon over I-plattform. Det oppsto da flere feilsituasjoner som man ikke hadde tatt høyde for. Eksempelvis så ble transaksjoner satt i gang i fagsystemet mens brevene ble overført til, og feilene oppsto i, et annet system. Man måtte da spesifisere feilmeldinger og følge opp brevene mellom to systemer der man tidligere hadde hele flyten i én silo. Utviklerne uttalte at denne problemstillingen hadde ført til mye hodebry under utviklingen.

I tillegg opplevde man utfordringer i forhold til at man ikke visste hvem som hadde ansvaret for innholdet på den andre siden, spesielt etter omorganiseringen av SITS.

Skatteetaten har 120-130 applikasjoner i sin systemportefølje fra før av. Gjennom introduksjon av nye fellestjenester som går på tvers av disse applikasjonene og arkitekturerne så har dette bydd på utfordringer. Figur 20 viser et forenklet bilde av porteføljen fra 2009. Bildet er naturligvis noe annerledes i dag, men det illustrerer like fullt kompleksiteten i Skatteetatens portefølje. Konteksten rundt Skatteetatens IT-arkitektur kan minne om den som fremstilles i Baskervilles (2010) beskrivelse av arkitekturen i bankene de studerte. Denne arven som man må dra med seg inn i SOA må man leve med, og ta hensyn til.



Figur 20. Forenklet bilde av Skatteetatens portefølje (Staverløkk, 2009)

6.3.10 Oppsummering av sentrale utfordringer

Studien viser at det har vært en del utfordringer relatert til Dokumentprosjektet. Ut fra en fortolkning i et sosio-teknisk perspektiv mener jeg at hovedårsaken til disse utfordringene stammer fra den samlede kompleksiteten i prosjektet. Dokumentprosjektet har blitt gjennomført som et systemutviklingsprosjekt som skal innføre en ny dokumentløsning gjennom en felleskomponent og flere fagkomponenter. Dette har skjedd i en kontekst der Skatteetaten nylig har innført flere nye plattformer (byggeblokker), men man har enda ikke utviklet retningslinjer og hentet erfaringer fra bruk (organisatoriske byggeblokker) fra disse.

Basert på resultatene fra studien er funnene delt inn i 6 hovedpunkter oppsummert i tabell 7.

Tabell 7 Sentrale funn i studien

Utfordring:	Beskrivelse:
Kompetanse	Med kompetanse menes den kompetansen som ligger i prosjektgruppene. Dette innbefatter kompetanse på systemutviklingsspråk, systemutviklingsmetode, tjenesteorientering og forretningsforståelse
Koordinering	Med koordinering menes koordinering og styring som er gjort innad i og mellom prosjektene
Kommunikasjon	Med kommunikasjon menes kommunikasjonen som har foregått både innad i og mellom prosjektene
Parallele prosjekter	Med parallelle prosjekter menes at man må se på følgene av at flere avhengige prosjekter har blitt gjennomført overlappende eller parallelt
Konsulentbruk	Med konsulentbruk menes det at vi må se på de konsulentene som har vært benyttet i prosjektene og hva de har tilført Skatteetaten
Prosjekt- og systemutviklingsmetodikk	Med prosjektmetodikk menes måten man har gjennomført prosjektet systemutviklingsmessig og ledelsesmessig

7. Diskusjon

Dokumentprosjektet er et viktig signalprosjekt for Skatteetaten da det innførte den første virkelige fellestjenesten i deres SOA-program. For å forstå Dokumentprosjektet og hvilke faktorer som påvirket det underveis må man ta hensyn til kompleksiteten i Skatteetatens IT-infrastruktur og konteksten prosjektet ble innført i. I denne studien har jeg valgt å benytte et sosio-teknisk perspektiv for å kunne forstå hvilke faktorer som har påvirket prosjektene underveis.

Studien har sett på hvordan Dokumentprosjektet har blitt gjennomført i Skatteetaten. Det synes som om det har vært forsket lite på SOA Governance og på hvordan man styrer konkrete SOA-prosjekter (Joachim, 2011). Jeg ønsket derfor å følge et prosjekt som skulle innføre en felleskomponent og studere hvilke faktorer som påvirker utfallet og hvilke utfordringer man får når man styrer et slikt prosjekt.

Basert på studien er det identifisert seks hovedkategorier som har påvirket Dokumentprosjektet i sin helhet. Disse er:

- Kompetanse
- Koordinering
- Kommunikasjon
- Parallele prosjekter
- Konsulentbruk
- Prosjekt- og systemutviklingsmetode

Kompetanse

Gjennom intervjuene med sentrale deltakere involvert i Dokumentprosjektet, har det kommet klart frem at kompetanse i prosjektgruppene har vært en utfordring underveis. Systemutviklerne i Skatteetaten har historisk sett gjennomført systemutvikling mot egne silo-baserte fagsystemer ved å bruke ulike systemutviklingsspråk. I Dokumentprosjektet skulle systemutviklingen gjøres i en tjenesteorientert infrastruktur. Det skulle benyttes en felles integrasjonsplattform og en felles systemutviklingsplattform som baserer seg på Java, et ukjent systemutviklingsspråk for de fleste systemutviklerne i Skatteetaten. Både integrasjonsplattformen og systemutviklingsplattformen var nylig innført slik at man ikke hadde laget retningslinjer for bruk eller dannet seg erfaringer på hvordan disse skulle benyttes.

Cockburn (2007) uttaler at et velfungerende team satt sammen av gode nok folk klarer å gjennomføre et prosjekt nesten uansett hvilken prosess eller teknologi som benyttes, men dette forutsetter nødvendigvis at man kan teknologien og systemutviklingsspråket. Det synes som om manglende kompetanse og erfaringer med systemutviklingsspråket har skapt en del utfordringer i prosjektet. Det er tungt å skulle lære seg et nytt systemutviklingsspråk samtidig som man skal gjennomføre et komplisert og distribuert systemutviklingsprosjekt.

Skatteetaten fulgte rådet fra blant andre Cockburn (ibid) om å tilføre prosjektgruppen kompetanse innen systemutvikling. Man startet også et utdanningsløp for systemutviklerne i henhold til Erls (2011) pilarer for SOA governance. En del av utfordringene i prosjektet skyldtes at man lå litt i forkant

vedrørende selve systemutviklingen slik at utvikling av retningslinjer og erfaringer fra plattformene kom i etterkant.

Ross (2003) anbefaler at man i en modulær struktur i hvert fall fokuserer på å ha egne arkitekter som kjenner organisasjonen og kan gi historikk når prosjektene går over til drift. Skatteetaten leide inn en del konsulenter som la sentrale føringer for Dokumentprosjektet. Dette kan ha påvirket løsningene som ble valgt. Man bør også passe på å hente ut erfaringene som er blitt gjort i prosjektet før konsulentene går ut av organisasjonen.

Konsulentene som ble leid inn i sentrale roller - spesielt i FDOK - manglet den nødvendige forretningsforståelsen i Skatteetaten for å kunne skape gode fellestjenester (BEA, 2005; Oracle, 2010). Den manglende forretningsforståelsen førte i en del tilfeller til forsinkelser fordi man ikke hadde nok kjennskap til forretningslogikk og kontekst rundt systemutvikling i Skatteetaten. Disse forsinkelsene førte i sum til at man fikk større forsinkelser for Dokumentprosjektet sett under ett.

Skatteetaten har økonomi og ressurser til å kunne utvide prosjektet slik at man får gjennomført Dokumentprosjektet selv om det har vært utfordringer og forsinkelser. Det er ofte slik at kostnadene ved innføring av SOA vil oppleves som høye, men at effekten trer frem over tid (BEA, 2005). Hadde ikke Skatteetaten vært i en så sterk situasjon ville man muligens vært nødt til å redusere prosjektets ambisjoner eller i verste fall kansellere det.

Koordinering

Dokumentprosjektet startet opprinnelig opp som fire enkeltprosjekter. FDOK ble oppfattet som navet blant prosjektene, men man manglet en felles styringsgruppe og et nødvendig mandat for å kunne ta avgjørelser.

Deltakerne i studien peker på at man satte for stor lit til kravspesifikasjon og kontrakter mellom tjenestene. Etter at kravspesifikasjonen var utviklet var det veldig lite koordinering mellom de ulike prosjektene. Dette førte til at informasjon om fremdrift i prosjektene og avhengigheter mellom dem ikke kom frem til systemutviklerne på begge sider. Både Erl (2011) og Cohn (2010) har fokusert på viktigheten av koordinering og samarbeid mellom prosjektgrupper for å sikre at tjenestene trekker i samme retning. I Dokumentprosjektet har manglende koordinering ført til flere utfordringer. Dette har igjen ført til forsinkelser i de ulike prosjektene og synes å ha vært en sterkt medvirkende årsak til at prosjektet ble forsinket med ett år.

Det er ikke en ukjent problemstilling at man må koordinere på tvers av ulike Scrumteam. Det er denne måten man skalere Scrum på. Det som i henhold til Cockburn (2007) da er beste praksis er at man i tillegg til daglige interne Scrummøter har «Scrum of Scrums» der én representant fra hvert team stiller og de koordinerer seg i mellom. De tre spørsmålene som der besvares vil gi teamene en god mulighet til å kunne koordinere seg i mellom, fange opp informasjon som trenger å flyte mellom prosjektene og synkronisere sprintene sine. Dette ble ikke gjort i de tidligste fasene av Dokumentprosjektet.

Etter at man formelt opprettet Dokumentprosjektet som en overbygging over de fire delprosjektene og innførte en egen styringsgruppe uttaler deltakerne at koordineringen har blitt forbedret. Dette har igjen ført til at prosjektene møter færre utfordringer.

Kommunikasjon

Kommunikasjon er et begrep som dukker opp ofte i litteraturen både i forhold til systemutvikling og i forhold til SOA (Cockburn, 2007; Cohn, 2010; Erl, 2011).

Dokumentprosjektet har blitt utviklet på flere ulike lokasjoner. Man har i alle prosjekter unntatt FDOK hatt dedikerte leveranser på hver lokasjon, slik at organiseringen ligner en multisite organisering (Cockburn, 2007).

Når man gjennomfører store systemutviklingsprosjekter, er kommunikasjon meget viktig, og slik har det også vært i Dokumentprosjektet. Både innad i prosjektene, men spesielt mellom prosjektene uttaler deltakerne at kommunikasjonen har vært for dårlig. Flere deltakere har uttalt at «lokasjon har noe å si» når de beskriver samarbeidsklimaet mellom prosjektene. Det synes som om det i stor grad er prosjektlederne som har kommunisert mellom prosjektene. På systemutviklernivå har det kun vært sporadisk kontakt på private initiativ. Uttalelser fra systemutviklerne tyder på at man har manglet den første kontakten som gjør at man neste gang tar opp telefonen og kontakter personen man vil ha tak i. Mange av elementene fra Cockburn (ibid) ville i så måte kunne vært tatt i bruk for å bedre denne kommunikasjonen. Da man gikk over fra å benytte en skreven kravspesifikasjon til å benytte brukerhistorier i større grad virker det som om FDOK fikk bedre oversikt over brukerkravene til tjenesten.

Cockburn (ibid) beskriver et «not-invented-here syndrome» der systemutviklerne vil utvikle løsningene sine selv i stedet for å implementere ferdige løsninger som andre har laget. Hadde man lagt bedre til rette for samarbeid og i tillegg pålagt systemutviklerne å samarbeide om å utvikle felles byggeblokker, ville man sannsynligvis kunne spart en del ressurser.

Kommunikasjonen fra FDOK til de andre prosjektene blir av deltakerne beskrevet som mangelfull. Det skapte flere utfordringer når FDOK ikke tok med seg deltakere fra de andre prosjektene i sentrale møter. Etter de sentrale møtene kommuniserte man heller ikke i tilstrekkelig grad beslutningene som ble tatt. Dette førte til at delprosjektene fikk vite om viktige rammefaktorer og endringer på sene tidspunkt i prosessen. Uttalelsen fra en av deltakerne er ganske beskrivende i så måte: *«Vi har sett at det er viktig at felleskomponenten lytter til innspill fra involverte parter og eksterne enheter og kommuniserer dette ut til alle deltakerne».*

Teamarbeid utgjør en av tre pilarer i Erls (2011) modell for SOA governance. Erl hevder at det er svært viktig med et balansert samarbeid både innad og mellom prosjektgrupper. I den sammenheng er det essensielt at man oppnår tillit mellom deltakerne i prosjektene slik at alle drar i samme retning. Manglende tillit kan i ytterste konsekvens føre til at aktørene jobber mot hverandre og at prosjektet feiler. I FDOK kan det synes som at man ikke har oppnådd god nok tillit og godt nok samarbeid mellom systemutviklerne i Oslo og Grimstad.

Parallelle prosjekter

Skatteetaten har i sitt moderniseringsprogram gjennomført mange prosjekter i løpet av de siste årene (se pt 4.3). Disse prosjektene har modernisert Skatteetatens IT-infrastruktur og IT-organisasjon. I samme omgang øker også kompleksiteten i konteksten som Dokumentprosjektet skal utvikles i.

Både Ross (2003) og Hirschheim et al. (2010) har forsket på modenhet innen ulike IT-arkitekturer. Begge anbefaler at man modnes gjennom modenhetsnivåene uten å gape over for mye. I Skatteetaten kan det virke som om de har gått for fort fram. For eksempel satte de i gang utvikling av store felleskomponenter før de hadde høstet erfaringer fra hvordan arkitekturen burde benyttes. Lee et al. (2010) understreker viktigheten av at man tar hensyn til kapasitet man tilgjengelig før man planlegger inkrementell innføring av tjenester. I Skatteetaten har man skapt en del tekniske byggeblokker i form av integrasjonsplattformen og systemutviklingsplattformen, men de organisatoriske byggeblokkene som skaper retningslinjene for bruken er enda ikke implementert før man går videre til neste nivå (BEA, 2005). BEA (ibid) anbefaler at man høster tjenester gjennom en inkrementell utvikling og gjenbraker funksjonalitet der det er mulig.

Omorganiseringen av SITS skapte en del organisatoriske utfordringer som man ikke hadde tatt høyde for. Dette førte til at Dokumentprosjektet etter omorganiseringen i enkelte tilfeller hadde problemer med å finne frem til hvilke personer som hadde ansvar for sentrale oppgaver.

Dokumentprosjektet selv består av fire prosjekter som har blitt utviklet parallelt. Dette har skapt en del utfordringer underveis. Systemutviklingen i hvert delprosjekt har blitt gjort mot en tjeneste i den andre enden som har vært i konstant endring. Når kommunikasjonen mellom prosjektene har vært mangelfull og kravspesifikasjonen ikke alltid ble fulgt så skapte det ytterligere utfordringer for prosjektene.

Konsulentbruk

Ross (2003) anbefaler at man skal ha interne systemarkitekter på prosjektene sine for å sikre at IT-infrastrukturen utvikles enhetlig og at man beholder kompetansen i egen organisasjon. Deltakere i Skatteetaten uttaler selv at de har en strategi om å ha internt personell på alle viktige roller som systemarkitekt og prosjektledere, men at de skal leie inn eksterne for å ta toppene av belastningen.

I Dokumentprosjektet leide Skatteetaten inn flere konsulenter. Dette har vært diskutert relatert til kompetanse og forretningsforståelse, men det fikk også følger de gangene man byttet konsulenter på prosjektene. De eksterne konsulentene hadde ikke erfaring fra Skatteetaten og det tok derfor lenger tid hver gang en konsulent ble byttet ut før vedkommende kunne bidra for fullt.

SOAs manifest (Arsanjani et al., 2009) sier at man skal fokusere på forretningsverdi foran teknisk strategi, og strategiske mål bør gå foran prosjekt-spesifikke fordeler. Dette kan være vanskeligere for en ekstern prosjektleder hvis ikke forretningsforståelsen er god nok og man blir målt på prosjektets leveranse i forhold til prosjektspesifikke mål.

Bruken av eksterne konsulenter synes i FDOK å ha fått innvirkning på tilliten internt i dette delprosjektet. Deltakerne uttalte at man aldri hadde noen felles oppstart i FDOK. Dette førte til at man ikke opparbeidet den nødvendige tilliten mellom personellet i Grimstad og personellet i Oslo.

Prosjekt- og systemutviklingsmetode

Deltakerne i studien uttaler at man benytter Scrum som systemutviklingsmetode og at man ikke tar noen spesielle forhåndsregler med tanke på prosjektmetodikk i Dokumentprosjektet. I en studie utført av Chow et al. (2008) ble det forsket på suksessfaktorer knyttet til smidig systemutvikling, og sentrale funn viste at en hensiktsmessig smidig utviklingsmetode var én av seks kritiske suksessfaktorer.

I smidige systemutviklingsmetoder (Cockburn, 2007) er det spesielt viktig at man fokuserer på kommunikasjonen i prosjektgruppene. Vi har sett at man her kunne gjort noen forbedringer. Spesielt kunne man hentet fordeler ved å gjennomføre «Scrum of Scrums» mellom prosjektgruppene for å oppnå bedre koordinering mellom prosjektene.

Det finnes lite forskning på ledelse og styring av konkrete SOA-prosjekter, men både rammeverkene (BEA, 2005; Oracle, 2007) og andre forfattere (Erl, 2011) understreker viktigheten av SOA governance i et SOA-program. Dette må også påvirke de enkelte prosjektene innad i SOA-programmet. For meg synes de som om prosjektlederne burde håndtert prosjektet som et spesielt og komplekst prosjekt, og hatt mer fokus på for eksempel kontroll og koordinering. Det er en forskjell mellom å gjennomføre et systemutviklingsprosjekt innad i en silo og det å utvikle fellestjenester og fagkomponenter på tvers av organisasjonen.

Tran et al. (2007) uttaler at kompleksiteten i en tjenesteorientert arkitektur blir så omfattende at man er nødt til å ta forhåndsregler for å få med hele konteksten rundt et systemutviklingsprosjekt. I prosjektet har man fått mange utfordringer som muligens kunne vært unngått hvis man hadde tatt høyde for kompleksiteten i styringen av prosjektet.

Skatteetaten burde etter erfaringene fra denne studien ta lærdom fra Dokumentprosjektet. Å utvikle en fellestjeneste på tvers av organisasjonen øker antall interessenter og øker også antall hensyn man må ta i prosjektet. I tillegg vil man ofte måtte utvikle tjenestekomponenter i flere forskjellige systemer samtidig med integrasjon mellom disse. Dette vil øke kompleksiteten i prosjektenes miljø og føre til at prosjektledere og systemutviklere må ha enda bedre kontroll, og kommunikasjon på tvers av organisasjonen når avgjørelser skal tas. Enkle grep som å innføre jevnlig møter mellom deltakere fra de ulike prosjektene har vist seg å ha en positiv effekt. Studien tyder på at det viktigste er at man innser at kompleksiteten i systemutviklingsmiljøet har en innvirkning på prosjektet, og at man derfor bør ta høyde for det når man velger prosjektmetodikk og systemutviklingsmetodikk.

Ved å ta høyde for kompleksiteten og endre måten prosjektene ble styrt på ville man kunnet unngå mange av de utfordringene som oppstår i prosjektet. Studien har identifisert utfordringer relatert til kompetanse, koordinering, kommunikasjon, parallelle prosjekter, konsulentbruk og prosjektmetodikk. Ved å fokusere på disse områdene i en tidlig fase og benytte rådene fra forskere og rammeverk vil man minimere sannsynligheten for at lignende utfordringer oppstår i senere prosjekter av lignende karakter.

8. Konklusjon og implikasjoner

Her presenteres konklusjon og implikasjoner fra studien. Studien har søkt å undersøke hvordan SOA-prosjekter gjennomføres i praksis og hvilke sentrale utfordringer man vil møte ved innføring av en fellestjeneste i en tjenesteorientert arkitektur. Basert på disse funnene søker studien så å identifisere hvordan en organisasjon kan ta lærdom av disse utfordringene ved innføring av fremtidige fellestjenester.

8.1 Konklusjon

Dokumentprosjektet er Skatteetatens første virkelige fellestjeneste på veien mot en tjenesteorientert organisasjon. Prosjektet fremstår derfor som et viktig *signalprosjekt* i Skatteetatens SOA-program. Flere deltakere har sagt at Dokumentprosjektet i den tidlige fasen ikke har vært noe mønsterprosjekt for Skatteetaten i form av at de har hatt en del utfordringer, og at tidsfrister og budsjetter har blitt overskredet. I den senere tid later det til at man har fått prosjektet på riktig kjøll og at man er i ferd med å levere en tjeneste som støtter godt opp under Skatteetatens dokumentbehov. Studien viser at den samlede kompleksiteten i Skatteetatens IT-infrastruktur og konteksten fellestjenesten skal implementeres i er en sentral årsak til at disse utfordringene oppsto.

Ved å studere Dokumentprosjektet fra et sosio-teknisk perspektiv og intervju sentrale deltakere i prosjektet, har det gitt meg anledning til å studere hvilke faktorer som påvirker utfallet ved utvikling av en fellestjeneste i en kompleks systemutviklingskontekst. Gjennom studien dannet det seg et bilde som gjorde det mulig å identifisere seks hovedkategorier som synes å være hovedårsakene til prosjektets utfordringer underveis. Disse kategoriene er 1) kompetanse, 2) koordinering, 3) kommunikasjon, 4) parallelle prosjekter, 5) konsulentbruk og 6) prosjekt- og systemutviklingsmetode.

Kompleksitet i systemutviklingsmiljøet vil alltid være utfordrende. Studien har vist at i et prosjekt som skal innføre en fellestjeneste i en tjenesteorientert arkitektur vil konteksten rundt prosjektet bli kompleks. Hvis man i prosjektledelsen har et økt fokus på de seks kategoriene som studien identifiserer og helheten i leveransen så vil man kunne redusere risikoen for at kompleksiteten medfører at prosjekter mislykkes i framtiden.

8.2 Implikasjoner for praksis

Mitt forslag til Skatteetaten er at man tar lærdom av Dokumentprosjektet. Det kan synes som at man har vært litt ivrige med innføringen av nye prosjekter. Etaten burde i større grad ha tatt seg mer tid til å lære av og høste erfaringer fra de nye plattformene som er innført før man startet nye prosjekter som bygger videre på denne funksjonaliteten. Det er gjort mye riktig ved å hente inn eksterne konsulenter med viktig kompetanse (Cockburn, 2007), men helheten i systemutviklingsmiljøet synes å ha blitt litt for komplekst på litt for kort tid.

Studien identifiserer koordinering mellom utviklingsteam som en sentral utfordring. For å bedre dette kan man innføre overordnede styringsgrupper og benytte Cohns (2010) «Scrum of Scrums» møter der representanter for hvert team møtes to til tre ganger i uken for å diskutere utfordringer og implikasjoner mellom prosjektene.

Det finnes lite forskning på ledelse og styring av konkrete SOA-prosjekter, men studiens funn tyder på at prosjektledere bør håndtere et prosjekt som innfører en fellestjeneste som et litt spesielt

prosjekt og ha mer fokus på for eksempel kontroll og koordinering. Det er en forskjell mellom å gjennomføre et systemutviklingsprosjekt innad i en silo og det å utvikle fellestjenester og fagkomponenter på tvers av organisasjonen. Prosjektledere som innfører fellestjenester bør fokusere på kompleksitet i utviklingskonteksten og ha et helhetlig perspektiv på tjenestene.

8.3 Implikasjoner for forskning

Studien gjør interessante funn relatert til innføring av fellestjenester i en tjenesteorientert arkitektur. Dette er et område hvor det foreløpig er gjort lite forskning. Forskningsmiljøet bør følge opp konkrete systemutviklingsprosjekter i SOA og kompleksitet i systemutviklingskontekst for å bygge mer kunnskap rundt hvordan disse kan gjennomføres og styres på en best mulig måte.

Denne studien er en singel casestudie og man trenger flere studier for å kunne undersøke om studiens funn kan generaliseres.

Studien har fokusert på en spennende retning – overgangen fra et overordnet SOA-program til innføringen av konkrete fellestjenester. Studien identifiserer at dette kan være et utfordrende og spennende område hvor man trenger mer kunnskap.

8.4 Begrensninger ved studien og forslag til videre forskning

Studien er en singel casestudie. Det er nødvendig med mer forskning på området for å utdype funnene fra denne studien. Det kunne vært interessant å utvide studien i Skatteetaten da dette er en stor organisasjon som har mange prosjekter og avdelinger. Det kunne også vært interessant å gjøre tilsvarende studier i andre organisasjoner og undersøke om man kom frem til de samme funnene.

Det er foreløpig gjennomført ganske lite forskning på SOA governance (Joachim, 2011). Dette er et område hvor man burde gjennomført flere studier for å øke forståelsen for hvordan dette bør gjennomføres.

Intervjuene ble gjennomført som semistrukturerte intervjuer. Selv om jeg prøvde å stille åpne spørsmål og hadde kontrollspørsmål hvor jeg spurte om det var noe annet deltakere ville legge til av informasjon, så kan introduksjonen av nye faktorer ha blitt skadelidende. Det er derfor en mulighet for at det finnes faktorer som har påvirket utfallet som jeg ikke har avdekket i denne oppgaven. Når det er sagt så er jeg fornøyd med intervjuene og føler at jeg har tatt de nødvendige forholdsregler for å legge til rette for å kunne avdekke de mest sentrale utfordringene i løpet av studiet.

På grunn av tidspunktet intervjuene ble gjennomført på, så fikk jeg ikke mulighet til å intervju den første prosjektlederen i FDOK. Dette var en ekstern konsulent som var innleid som prosjektleder på dette prosjektet. Vedkommende kunne sannsynligvis hatt nyttig tilleggsinformasjon om gangen i prosjektet som igjen kunne gitt studien mer rike data. Likevel så har jeg intervjuet mange sentrale aktører i prosjektene som hadde praktisk kunnskap og erfaringer fra prosjektprosessen slik at jeg føler at summen av deres observasjoner gir et helhetlig bilde.

9. Referanser

- Arsanjani, A., Booch, G., Boubez, T., Brown, P. C., Chappell, D., deVadoss, J., Erl, T., Josuttis, N., Krafzig, D., Little, M., Loesgen, B., Thomas Manes, A., McKendrick, J., Ross-Talbot, S., Tilkov, S., Utschig- Utschig, C., and Wilhelmssen, H. (2009). *SOA Manifesto*. City. [Online] Tilgjengelig: <http://www.soa-manifesto.org/> [05.01.2012]
- Avgerou, C. (2001). The Significance of Context in Information Systems and Organizational Change. *Information Systems Journal*, Vol. 11, s. 43–63
- Baskerville, R.L., Cavallari, M., Hjort-Madsen, K., Pries-Heje, J., Sorrentino, M. and Virili, F. (2010). The Strategic Value of SOA: A Comparative Case Study in the Banking Sector. *Int. J. Information Technology and Management*, Vol. 9, No. 1, s. 30-53
- BEA White Paper. (2009). BEA's SOA Reference Architecture; A Foundation for Business Agility. [Online] Tilgjengelig: http://rasea.googlecode.com/svn-history/r44/doc/trunk/referencia/SOA_Ref_Arch_wp.pdf [05.01.2012]
- BEA White Paper. (2005). Domain Model For SOA; Realizing the Business Benefit of Service-Oriented Architecture. [Online] Tilgjengelig: http://www.soablueprint.com/yahoo_site_admin/assets/docs/BEA_SOA_Domains_WP.290214359.pdf [05.01.2012]
- Beimborn, D., Joachim, N., and Muenstermann, B. (2009) Impact of Service-Oriented Architectures (SOA) on Business Process Standardization - Proposing a Research Model, *ECIS 2009 Proceedings*
- Benbasat, I., Goldstein, D.K. and Mead, M. (1987). The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, Vol. 11, No. 3, s. 369-386.
- Bygstad, B. og Munkvold, B.E., (2007) The Significance of Member Validation in Qualitative Analysis: Experiences from a Longitudinal Case Study, *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, s. 243b.
- Chow, T., and Cao, D.-B. (2008). A Survey Study of Critical Success Factors in Agile Software Projects. *The Journal of Systems and Software*, Vol. 81, No. 6, s. 961-971.
- Cockburn, A. (2007). *Agile Software Development: The Cooperative Game*. Boston: Addison-Wesley.
- Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Upper Saddle River, N.J.: Addison-Wesley
- Creswell, J. (1998). *Qualitative Inquiry and Research Design: Choosing among Five Traditions*. Thousand Oaks, CA: Sage
- Creswell, J.W. (2009): *Research Design, Qualitative, Quantitative, and Mixed Methods Approaches*. Los Angeles: Sage.
- Davenport, T.H. and Short, J.E. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, Vol. 31, No. 4, s. 11-27

Eisenhardt, K.M. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, Vol. 14, No. 4, s. 532-550.

Erl, T. (2011). *SOA Governance: Governing Shared Services On-Premise and in the Cloud*. Boston: Prentice Hall.

Fallmyr, F. og Bygstad, B. (2011). Enterprise Architecture to Enhance Organizational Agility? An Exploratory Study. I Fallmyr et al. (Red.), *Proceedings of NOKOBIT 2011*, Tromsø, Tapir Akademiske Forlag, s. 209-220.

Fowler, M. & Highsmith, J. (2001) The Agile Manifesto. Software Development. [Online] Tilgjengelig: <http://agilemanifesto.org/>. [12.05.2012]

Hirschheim, R., Welke, R., and Schwarz, A. (2010). Service-Oriented Architecture: Myths, Realities, and a Maturity Model. *MIS Quarterly Executive*, Vol. 9, No. 1, s. 37-48.

Hustad, E. og Staverløkk, L. (2013). Implementing a Service-Oriented Architecture: A Technochange Approach. Pooley, R.J., Coady, J., Linger, H., Barry, C., Lang, M., and Schneider, C.(Red.) *Information Systems Development: Reflections, Challenges and New Directions*, Springer Verlag. Tilgjengelig 31. Januar 2013.

ISACA. (2012). COBIT. [Online] Tilgjengelig: <http://www.isaca.org/Knowledge-Center/COBIT/Pages/Overview.aspx>. [21.04.2012]

Jacobsen, D.I. (2005). *Hvordan Gjennomføre Spørreundersøkelser?— Innføring i Samfunnsvitenskapelig Metode*. 2 utgave. Høyskoleforlaget. Kristiansand.

Joachim, N. (2011). A Literature Review of Research on Service-Oriented Architectures (SOA): Characteristics, Adoption Determinants, Governance Mechanisms, and Business Impact. *AMCIS 2011 Proceedings - All Submissions*. Paper 339.

Kaplan, R.S. & Norton, D.P. (1992). The Balanced Scorecard – Measures that drive performance. *Harvard Business Review*, Januar-Februar, s. 70-80.

Klein H. K. and Myers M.D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly*, Vol. 23, No. 1, s. 67-93

Lee, J.H., Shim, H-J., Kim, K.K. (2010). Critical Success Factors in SOA Implementation: An Exploratory Study, *Information Systems Management*, Vol. 27, No. 2, s. 123-145

Liegl, P. (2007). The Strategic Impact of Service Oriented Architectures. *Paper presented at the Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, Tucson, Arizona, USA.

Marks, E. A., & Bell, M. (2006). *Service-Oriented Architecture: A Planning and Implementation Guide for Business and Technology*. Hoboken, New Jersey: John Wiley & Sons.

Markus, M.L. (2004). Technochange management: using IT to drive organizational change. *Journal of Information Technology*, Vol. 19, s. 3-19.

Miles, M.B., Huberman, A.M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. Thousand Oaks: Sage

Munkvold, B.E. (2000) Tracing the Roots: The Influence of Sociotechnical Principles on Modern Organisational Change Practices. In Coakes, E., Lloyd-Jones, R. & Willis, D. (eds.), *The New Sociotech: Graffiti on the Long Wall*. Springer-Verlag, London, s. 13-25.

OASIS. (2006). Reference Model for Service Oriented Architecture 1.0. [Online] Tilgjengelig: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html> [14.05.2012]

Oracle (2007). SOA Governance: Framework and Best Practices. [Online] Tilgjengelig: <http://www.oracle.com/us/technologies/soa/oracle-soa-governance-best-practice-066427.pdf>. [05.01.2012]

Oracle (2008). Oracle and BEA. [Online] Tilgjengelig: <http://www.oracle.com/us/corporate/acquisitions/bea/index.html>. [05.01.2012]

Oracle (2010). SOA Maturity Model – Guiding and Accelerating SOA Success. [Online] Tilgjengelig: <http://www.oracle.com/technetwork/topics/entarch/oracle-wp-soa-maturity-model-176717.pdf>. [05.01.2012]

Ordanini, A., & Pasini, P. (2008). Service Co-Production and Value Co-Creation: The Case for a Service-Oriented Architecture (SOA). *European Management Journal*, Vol. 6, s. 289–297.

Porter, M. E. (1985). *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press. New York.

Ross, J.W. (2003). Creating a strategic it architecture competency: Learning in stages. *MIS Quarterly*, Vol. 2, No. 1, s. 31-43.

Skatteetaten. (2012). [Online] Tilgjengelig: <http://www.skatteetaten.no>. [15.06.2012]

Staverløkk, L. (2009). Kritiske suksessfaktorer og utfordringer ved innføring av tjenesteorientert arkitektur: En casestudie i Skatteetaten. Masteroppgave ved Universitetet i Agder

Stenerud, A., & Thorvaldsen, L. (2012) Ny Altinn-skandale. *Dagbladet*. [Online] Tilgjengelig: <http://www.dagbladet.no/2012/03/20/nyheter/innenriks/altinn/selvangivelsen/20770039/>. [15.06.2012]

Tran, H., Zdun, U., Dustdar, S. (2007). View-based and Model-driven Approach for Reducing the Development Complexity in Process-driven SOA. *Proceedings of the International Working Conference on Business Process and Services Computing (BPSC)*, s. 105-124, Gesellschaft für Informatik

Walsham, G. (1995) Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information Systems*, Vol. 4, No. 2, s. 74-81.

Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, Vol. 15, s. 320–330

Weill, P., Subramani, M.R., and Broadbent, M. (2002) Building IT Infrastructure for Strategic Agility. *Sloan Management Review*, Vol. 44, No. 1, s. 57-65.

Weill, P. (2004). Don't just lead, govern: How top-performing firms govern IT. *MIS Quarterly*, Vol. 3, No. 1, s. 1-17

Yin, R.K. (1984). *Case Study Research, Design and Methods.*, Beverly Hills, California. Sage Publications