

**UNIVERSITETET I OSLO**  
**Institutt for informatikk**

**Retningslinjer for  
prosessforbedring  
med vekt på  
endringshåndtering  
i webutviklings-  
prosjekter**

Anette Cecilie Lien

**Hovedfagsoppgave**

1. februar 2002





# Sammendrag

Det overordnede temaet i oppgaven er endringshåndtering i evolusjonære webutviklingsprosjekter. Evolusjonær utvikling kan karakteriseres ved at man har uferdige inkrementelle leveranser som utvikler seg mot å bli et ferdig produkt. Hvert inkrement kan leveres til sluttbrukerne for tilbakemeldinger. En evolusjonær prosessmodell er mottakelig for endringsforslag fra utviklerorganisasjon, kunde og brukere underveis i utviklingen. Målet er blant annet økt tilfredshet med produktet fra brukerne, samt økt risikohåndtering basert på brukerrespons. Endringshåndtering er et felt som griper inn i alle deler av utviklingsprosessen; fra kravspesifikasjon til ferdig leveranse. For å gi endringshåndteringen en kontekst, beskrives det først i oppgaven generelle erfaringer fra evolusjonære webutviklingsprosjekter, før det fokuseres spesifikt på endringshåndtering.

Oppgaven består av:

- En erfaringsrapport basert på intervjuer med 15 deltagere i tre evolusjonære webutviklingsprosjekter. Et litteratursøk gjort i tilknytning til denne oppgaven viste at det ikke er utført noen tilsvarende erfaringsinnhenting tidligere.
- Et sett med retningslinjer for evolusjonær utvikling basert på erfaringene
- En prosessbeskrivelse for endringshåndtering basert på retningslinjene
- En kjørbare prototype på et verktøy for endringshåndtering basert på prosessbeskrivelsen. Verktøyprototypen tar hensyn til det økonomiske perspektivet rundt endringshåndtering, samt støtter kommunikasjonen mellom sluttbrukere, kunde og utviklere på en måte som skiller prototypen fra tilsvarende verktøy for endringshåndtering på markedet.

Resultatene fra erfaringsinnhentingene viste at det kan det være en utfordring å håndtere endringsønsker som dukker opp underveis i utviklingen på en strukturert og kostnadseffektiv måte, og viste et klart behov for en mer formalisert måte å takle endringshåndteringen på. Dette ga opphav til utviklingen av prosessbeskrivelsen og prototypen på et verktøy som støtter endringshåndtering.



# Forord

Denne oppgaven er skrevet som en del av Cand.Scient.-graden ved avdeling for Industriell systemutvikling ved Institutt for informatikk, Universitetet i Oslo. Arbeidet med oppgaven har pågått i tidsrommet august 2000 - januar 2002.

Denne oppgaven er et resultat av et samarbeid mellom veileder Erik Arisholm ved Institutt for informatikk, Jon Skandsen fra bedriften Genera og meg selv.

Jeg vil gjerne takke min inspirerende veileder Erik Arisholm for støtte, engasjement og god veiledning gjennom hovedfaget. Det betød også veldig mye for meg at han støttet meg gjennom presentasjonen av artikkelen i Irland. Jeg vil videre rette en takk til alle som ble intervjuet — for at de tok seg tid og for at de korrigerer intervjuanalysene i etterkant. Dessuten fortjener de fem testbrukerne en hilsen. Videre er det naturlig å takke Genera for at de ga meg tilgang til ressurser i bedriften. Utviklingsleder Jon Skandsens tid og hjelp har gjort mye for resultatene i denne oppgaven. Jeg er også takknemlig for verdifulle kommentarer til oppgaven fra Dag Sjøberg, Silje Grid Stakston og Bjarne Johannessen.

Tusen takk til alle mine morsomme venner på Ifi og Genera for å ha lyst opp hverdagen! Det som spesielt har betydd mye for meg underveis er Siljes gode humør, de koselige kafébesøkene med Merete og Siri, søte postkort fra Kari, at Marianne kom ofte til Skrim og Friends-kveldene hos min gode venn Arne Meyer. Til slutt vil jeg takke familien min; mor, far og Jørgen for at de alltid er der for meg — uansett.



# Innhold

<b>1</b>	<b>Introduksjon</b>	<b>1</b>
1.1	PROFIT-prosjektet . . . . .	1
1.2	Genova . . . . .	3
1.3	Fokus i oppgaven . . . . .	4
1.4	Endringshåndtering . . . . .	4
1.5	Problemstilling . . . . .	5
1.6	Bidrag . . . . .	6
1.7	Språkbruk . . . . .	7
1.8	Metode . . . . .	7
1.9	Avgrensning . . . . .	10
1.10	Oppbygning av oppgaven . . . . .	10
<b>2</b>	<b>Evolusjonær utvikling av webapplikasjoner</b>	<b>11</b>
2.1	Endringer underveis i systemutviklingen . . . . .	11
2.2	Formalisering av endringshåndteringen . . . . .	12
2.3	Prosessmodeller . . . . .	15
2.4	Ulike evolusjonære, inkrementelle prosessmodeller . . . . .	23
2.5	Avsluttende bemerkninger . . . . .	27
<b>3</b>	<b>Metode</b>	<b>29</b>
3.1	Innhenting av materiale . . . . .	29
3.2	Metodevalg . . . . .	33
<b>4</b>	<b>Erfaringer fra webutviklingsprosjektene</b>	<b>39</b>
4.1	TelMont . . . . .	40
4.2	BibTid . . . . .	49
4.3	UniBase . . . . .	53
4.4	Retningslinjer for prosess . . . . .	56
4.5	Relatert arbeid . . . . .	62
4.6	Mulige feilkilder . . . . .	68
<b>5</b>	<b>En verktøyprototype for endringshåndtering</b>	<b>73</b>
5.1	Motivasjonen for å lage verktøyprototypen . . . . .	74
5.2	Prosessbeskrivelse . . . . .	75

5.3	Funksjonalitet og utseende ved verktøyprototypen . . . . .	79
5.4	Utvikling og evaluering av verktøyprototypen . . . . .	84
5.5	Sammenligning med andre verktøy . . . . .	86
5.6	Avsluttende bemerkninger . . . . .	92
<b>6</b>	<b>Konklusjon</b>	<b>93</b>
6.1	Erfaringsinnhenting og retningslinjene for prosessforbedring	93
6.2	Prosessbeskrivelsen og verktøyprototypen for endringshåndtering . . . . .	95
6.3	Videre arbeid . . . . .	95
<b>A</b>	<b>Evolutionary Development of Web-applications — Lessons Learned</b>	<b>103</b>
A.1	Introduction . . . . .	104
A.2	Improving the Genova Process . . . . .	104
A.3	Experiences from the Web Development Projects . . . . .	107
A.4	Process Guidelines . . . . .	111
A.5	Related Work . . . . .	114
A.6	Conclusions and Future Work . . . . .	115
<b>B</b>	<b>Intervjuguide</b>	<b>117</b>
<b>C</b>	<b>Transkribert materiale</b>	<b>121</b>
C.1	Intervju med prosjektleder hos kunden på TelMont . . . . .	121
C.2	Intervju med prosjektleder i utviklerorganisasjonen på TelMont	141
C.3	Intervju med utvikleren “B” på TelMont . . . . .	156
C.4	Intervju med utvikleren “K” på TelMont . . . . .	174
<b>D</b>	<b>Spørsmål til e-postintervjuene med brukerne på TelMont</b>	<b>189</b>
D.1	Intervjuspørsmål: . . . . .	189
<b>E</b>	<b>Svar fra e-postintervjuene med brukerne på TelMont</b>	<b>191</b>
E.1	Svar fra TelMont-bruker 1 . . . . .	191
E.2	Svar fra TelMont-bruker 2 . . . . .	193
E.3	Svar fra TelMont-bruker 3 . . . . .	195
E.4	Svar fra TelMont-bruker 4 . . . . .	196



# Figurer

1.1	Forholdet mellom denne hovedfagsoppgaven, Genova web-prosess og Genova-prosessen . . . . .	2
1.2	Metodiske faser i resultatinnsamlingen . . . . .	8
2.1	Sommervilles forslag til en prosess for endringshåndtering . .	13
2.2	Kategorisering av ulike prosessmodeller . . . . .	15
2.3	Fossefallsmodellen . . . . .	16
2.4	Inkrementell utvikling . . . . .	18
2.5	Flytdiagram over Genovaprosessen . . . . .	26
5.1	Aktivitetsdiagram for endringshåndtering . . . . .	76
5.2	Tilstandsdiagram for et endringsønske . . . . .	78
5.3	Menyen på verktøyprototypens hovedside . . . . .	80
5.4	Skjema for å fylle inn nytt endringsønske . . . . .	80
5.5	Utsnitt av utviklernes logg . . . . .	81



# Tabeller

3.1	Oversikt over personene som ble intervjuet . . . . .	30
4.1	Oversikt over de tre webutviklingsprosjektene . . . . .	40
4.2	Oversikt over erfaringer fra prosjektet TelMont . . . . .	41
4.3	Oversikt over erfaringer fra prosjektet BibTid . . . . .	50
4.4	Oversikt over erfaringer fra prosjektet UniBase . . . . .	54
5.1	Databaseskjemaet som ligger til grunn for verktøyprototypen	83
A.1	Overview of the interviewees . . . . .	105
A.2	Overview of the studied development projects . . . . .	108



# Kapittel 1

## Introduksjon

Gjennom de siste årene har bedriften Genera AS vært involvert i mange webutviklingsprosjekter. Gjennom deltagelse i disse prosjektene har det blitt klart for bedriften at webutvikling har mange karakteristika som krever en egentilpasset prosess-støtte. De ønsker derfor å utvikle en prosessmodell som skal gi støtte for evolusjonær utvikling av webapplikasjoner. Denne prosessmodellen skal være en forlengelse av den evolusjonære modellen Genova-prosessen som bedriften tidligere har utviklet, og skal kalles Genova web-prosess.

### 1.1 PROFIT-prosjektet

Gjennom sitt arbeid med denne prosessforbedringen, er Genera blitt knyttet til forskningsprosjektet PROFIT, som er et forskningsprosjekt i regi av Norges Forskningsråd. Denne hovedoppgaven har inngått i dette forskningsprosjektet. PROFIT – *“PROcess improvement For IT industry”* - fokuserer på å utvikle metoder og retningslinjer for prosessforbedring med vekt på målinger og gjenbruk av erfaringer og kunnskap (Ulsund 2001). Det overordnede mål for PROFIT er *“å øke konkurransedyktigheten og lønnsomheten til norsk IT-industri gjennom å fokusere på læring og prosessforbedring i det kontinuerlig skiftende miljøet rundt programvare-utvikling”* (Ulsund 2001).

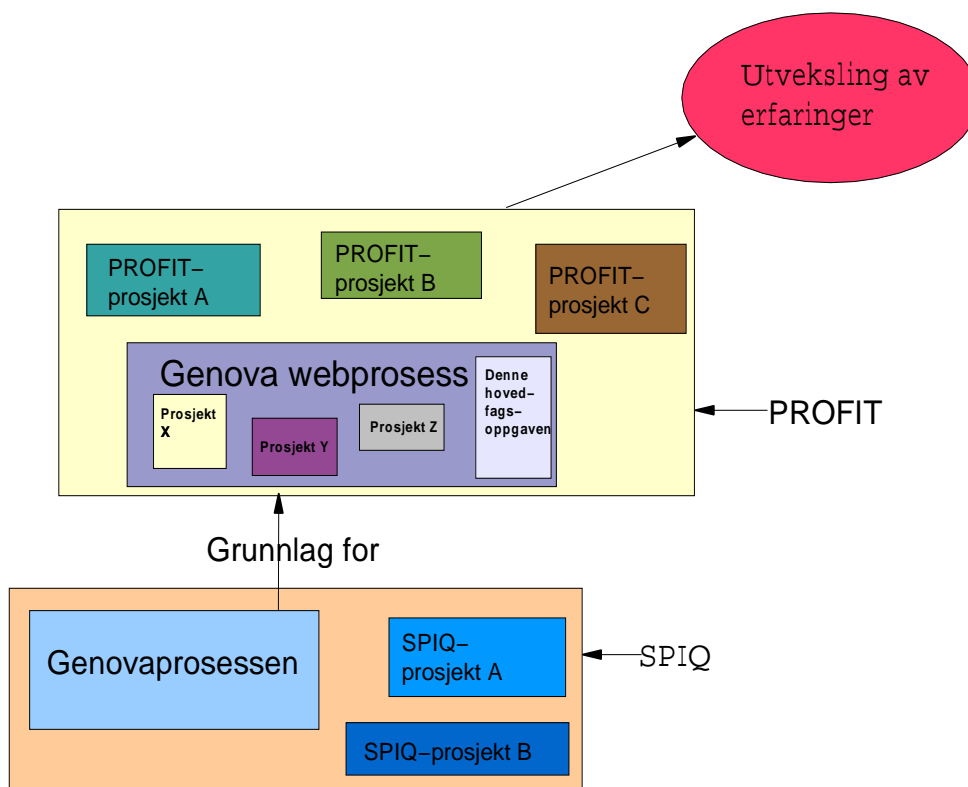
Fokusarena for PROFIT er prosessforbedring under usikre omstendigheter i et marked med hurtige endringer. Målet med forskningsprosjektet er erfaringsinnhenting fra forskjellige utviklingsprosjekter i ulike firmaer for å kunne gjenbruke erfaringer og kunnskap på tvers av bedriftene. Forskningsprogrammet PROFIT startet i mars 2000, og vil fortsette til ut år 2002. PROFIT er et samarbeidsprosjekt mellom forskere og næringsliv; en rekke bedrifter er involvert.<sup>1</sup> Alle bedriftene som er med i PROFIT skal i separate

---

<sup>1</sup>Samarbeidet foregår mellom Universitetet i Oslo, NTNU, SINTEF, Firm, Bravida Geomatikk, InfoStream, Kongsberg Spacotec, Tandberg Data, Ericsson, Icon Medialab, Kongsberg Defence Communication, mogul.com, TV2 Interaktiv og Genera.

delprosjekter utføre prosessforbedringstiltak, for så og møtes for å utveksle erfaringer.

Denne hovedoppgaven er knyttet opp mot et delprosjekt innen PROFIT som Genera og Institutt for Informatikk ved Universitetet i Oslo samarbeider om – utviklingen av Genova webprosess. Oppgaven gir innspill til Genova webprosess i form av generelle retningslinjer for prosessforbedring og en prosessbeskrivelse og verktøystøtte for endringshåndtering. Genova webprosess bygger på den allerede eksisterende evolusjonære prosessmodellen Genova-prosessen som ble utviklet av Genera under forskningsprosjektet SPIQ – “*Software Process Improvement for better Quality*”. SPIQ var et forskningsprosjekt i regi av Norges Forskningsråd. Prosjektet hadde 15 deltagende bedrifter og fokuserte på prosessforbedring og erfaringsutveksling på tilsvarende måte som PROFIT (Ulsund 2000). Slektskapet mellom SPIQ, PROFIT, Genova webprosess og denne oppgaven er skissert i figur 1.1.



Figur 1.1: Forholdet mellom denne hovedfagsoppgaven, Genova webprosess og Genova-prosessen

## 1.2 Genova

Genera utvikler og selger Genova, som er et avansert CASE-verktøy for objektorientert analyse og design, dialogmodellering og automatisk applikasjons- og databasegenerering (Arisholm, Skandsen, Sagli & Sjøberg 1999). Genova baserer seg på UML-modeller, er fullt integrert med Rational Rose og har en egentilpasset evolusjonær prosessmodell. Genova består av to deler:

- en prosessmodell (Genova-prosessen)
- et verktøy (Genova-verktøyet)

**Genova-prosessen** er med sin evolusjonære karakter tenkt som en effektiv måte å håndtere risiki på. Eksempler på risiki kan være innføring av ny teknologi (for eksempel webteknologi) eller upresise og skiftende krav (Boehm 1988). Prosessmodellen søker å unngå at man får et avvik mellom kravspesifikasjon og det systemet brukerne egentlig ønsker seg (Vogt 2001). Grunnideen er å finne usikkerhetsmomenter ved kravspesifikasjonen på et tidligst mulig stadium i prosjektet gjennom å utvikle systemet inkrementelt framfor å satse på den tradisjonelle fossefallstilnærmingen (Royce 1970). Genova-prosessen er tenkt som en nedskalert versjon av Rational Unified Process (RUP) (Arisholm, Skandsen, Sagli & Sjøberg 1999). Argumentet bak dette er at RUP er svært omfangsrik, og at en mindre omfattende evolusjonær prosess ville øke sjansen for suksess i mindre utviklingsprosjekter.

Genera ønsker nå å lage en **Genova webprosess** som et prosessforbedringstiltak og som et bidrag innen PROFIT. Genova webprosess skal være en utvidelse av Genovaprosessen som er tilpasset evolusjonær utvikling av webapplikasjoner. Målet er å utvikle retningslinjer for evolusjonær webutvikling. Retningslinjene skal baseres på en erfaringsinnhenting fra ulike webutviklingsprosjekter i Genera. Basert på erfaringsinnhenting skal det deretter defineres en Genova webprosess. Denne webutviklingsprosessen skulle så evalueres på nye prosjekter. Evalueringen er ment å vise hvorvidt retningslinjene ble brukt, og i hvilken grad de var nyttige.

**Genova-verktøyet** skal hjelpe utviklere med å generere brukergrensesnitt allerede under design- og analysefasen (Vogt 2001). Hovedaktivitetene i Genova-verktøyet er *dialogmodellering* og *databasemapping* basert på UML-modeller (Næss 2001). Genereringen skjer på grunnlag av Use Case-modeller man lager ut i fra kravspesifikasjonen og klassemodeller man tegner i et modelleringsverktøy som Rational Rose (Arisholm, Benestad, Skandsen & Fredhall 1998, Næss 2001). Dialogmodelleringen gjør at man kan velge ut deler av klassemodellen og hurtig lage et brukergrensesnitt ut i fra de klassene man velger. Databasemappingen gjør at man enkelt lager et database-

skjema ut i fra valgte klasser i klassemodellen. Brukergrensesnittet man genererer skal være en prototype på dialogvinduer. Disse kan man lett gjøre forandringer på i samarbeid med kunden. Hensikten med en slik rask prototyping i samarbeid med kunden, er å avdekke feiltolkninger og misforståelser i kravspesifikasjonen så tidlig som mulig i utviklingsprosessen. Man utvikler deretter applikasjonen basert på prototypene man genererte.

I sammenheng med utviklingen av Genova webprosess utvikles også funksjonalitet i Genova for utvikling av websider — Genova webgenerator (Næss 2001). Genova webgenerator søker å generere prototyper med grunnleggende funksjonalitet og design for webapplikasjoner ut i fra UML-modeller. Eksempler på ulike elementer som genereres er stylesheets, kode for server- og klientside, database-elementer og HTML-kode (inkludert radioknapper, flervalgskbokser, lister med mer).

Denne hovedoppgaven inneholder bidrag mot både prosessmodellen og verktøystøtten i Genova webprosess.

### 1.3 Fokus i oppgaven

Det overordnede temaet for denne oppgaven er endringshåndtering i evolusjonære webutviklingsprosjekter. Endringshåndtering er et felt som griper inn i alle deler av utviklingsprosessen — det være seg kravspesifikasjon, brukermedvirkning, forhold kunde/leverandør, prototyping eller estimering. Det synes derfor naturlig å beskrive også disse aspektene ved utviklingen for å gi en bedre forståelse av hvorfor endringshåndteringen hadde forbedringspotensiale. Oppgaven utarbeider derfor retningslinjer for prosessforbedring generelt, og fokuserer deretter på endringshåndtering spesielt gjennom en prosessbeskrivelse og prototype på et verktøy for endringshåndtering.

Forskjellene mellom webutviklingsprosjekter og tradisjonelle utviklingsprosjekter viskes i dag mer og mer ut. Det er en tendens til at tradisjonelle applikasjoner gjerne gis et brukergrensesnitt som er basert på webteknologi. Erfaringene som her ble høstet inn var fra prosjekter som hadde web som applikasjonsdomene. Selv om fokus for denne oppgaven er lagt på webutviklingsprosjekter, er det likevel rimelig å anta at mye av teorien vil ha store fellestrekk med utvikling av tradisjonelle applikasjoner.

### 1.4 Endringshåndtering

Årsaker til at programvaresystemer må endres kan være misforståtte, nye eller endrede krav fra brukeren, endrede lover og reguleringer, feil i design eller kode, tilpasninger til ny hardware og operativsystemer eller tilpasninger til andre programvaresystemer (Lientz, Swanson & Tompkins 1978). Endringer er en nødvendighet i store programvaresystemer fordi organisator-



iske behov og krav som regel forandrer seg under levetiden til et system (Sommerville 1995). Endringer i organisasjonens behov krever at tilsvarende endringer gjøres i programvaren.

Et *endringsønske* er en forespørsel om å endre en produkt del eller en produktprosess. En endringsforespørsel vil typisk dokumentere informasjon om hvem som har kommet med ønsket, en analyse av nåværende problem, en foreslått løsning og løsningens kostnader (Kruchten 2000). Det er som regel prosjektledelsen som tar for seg administrasjon av endringsforespørsler som forekommer underveis i utviklingsprosessen (Sjøberg 1993). Mottak av endringsforespørsler på et slikt nivå kalles vanligvis *endringshåndtering* (Humphrey 1989, Sjøberg 1993, Sommerville 1995). Endringshåndtering defineres av Kruchten som en aktivitet for å kontrollere og holde oversikten over endringer i utviklingsprosjekter (Kruchten 2000). I henhold til Sjøberg foregår endringshåndtering ideelt sett ved at alle endringer som kommer inn evalueres med hensyn på behovet for endring, hva endringen vil gjøre med systemet og prosjektet, tidsestimering med mer (Sjøberg 1993). Når man så gjennomfører en endring, samler man informasjon om hvem som gjorde hva når, hva statusen er, hva som gjenstår av arbeid og eventuelle andre punkter.

## 1.5 Problemstilling

Opgaven har to problemstillinger. Problemstilling 1 vektlegger et bredt fokus på generelle erfaringer fra evolusjonær webutvikling, mens problemstilling 2 snevrer inn fokus spesifikt mot endringshåndtering.

***Problemstilling 1:*** *Hvilke generelle erfaringer fra evolusjonær webutvikling kan være nyttige for andre evolusjonære webutviklingsprosjekter, og hvordan kan retningslinjene formaliseres i Genova webprosess?*

Besvarelsen av problemstilling 1 innebærer en erfaringsinnsamling fra tre webutviklingsprosjekter der Genera har vært involvert. Basert på de innsamlede erfaringene foreslås retningslinjer for evolusjonær webutvikling. Retningslinjene vil gi innspill til Genova webprosess og kan brukes av andre webutviklingsprosjekter.

Resultatene viste særlig at det å håndtere endringer i en evolusjonær utvikling kan være en utfordring. Dette kommer av at endringshåndtering bør sikre kommunikasjonen mellom sluttbrukere, kunde og utviklere på en slik måte at ingen av partene blir tilsidesatt. Ellers er det mindre sannsynlighet for at produktet blir som forventet for alle parter, det være seg forventninger rundt kostnader, brukervennlighet, nødvendig funksjonalitet eller annet. Resultatene indikerte også at endringshåndtering kan være vanskelig uten en formell prosess, fordi deler av nødvendig kommunikasjon kan bli forbigått, for eksempel ved at kunde ikke blir spurt til råds hvorvidt en foreslått endring bør tildeles de nødvendige økonomiske midler. Et problem kan også være

at brukerne kommer med så mange ønsker om endringer at endringshåndteringen kommer ut av kontroll fordi utviklerorganisasjonen mangler formelle rutiner for å håndtere endringsønskene. Å lage en prosessbeskrivelse og tilhørende verktøystøtte for endringshåndteringen i evolusjonære webutviklingsprosjekter kan være midler for å gjenvinne kontroll over noen av utfordringene rundt endringshåndtering. Dette gir opphav til problemstilling 2.

***Problemstilling 2:** Hvilken prosessbeskrivelse og hvilken verktøystøtte synes hensiktsmessig for endringshåndtering i evolusjonær webutvikling, og hvilke innspill gir dette til Genova-verktøyet?*

Intervjuene viste klart at mange deltagere etterlyste en formell prosess for endringshåndtering i prosjektene de deltok i fordi manglende håndtering av endringsforespørsler førte til en ukontrollert utvikling i prosjektene. Ut i fra dette synets det naturlig å utvikle en prosessbeskrivelse og en prototype på verktøystøtte for dette. Ønsket er at den formaliserte prosessen en verktøystøtte støtter opp under kan hjelpe med å strukturere endringshåndtering på webutviklingsprosjekter der aktørene (kunde, utviklere, brukere) kan være lokalisert på ulike geografiske plasseringer. Verktøyprototypen er situert på web for at den skal være enkelt tilgjengelig. Det er også foretatt en preliminær evaluering av verktøyet. Videre evaluering vil være neste steg i en iterativ syklus som foretas innen PROFIT-prosjektet. Prosessbeskrivelsen og prototypen er et innspill til Genova-verktøyet, og er planlagt integrert i Genova i nær framtid.

## 1.6 Bidrag

Denne oppgaven bidrar til PROFIT-prosjektet gjennom å gi innspill til Genova webprosess. Bidraget består av:

- Kartlegging av erfaringer fra tre evolusjonære webutviklingsprosjekter. Litteratursøket gjort i tilknytning til denne oppgaven viste at det ikke er utført noen tilsvarende erfaringsinnhenting fra evolusjonære webutviklingsprosjekter tidligere.
- Utarbeidelse av retningslinjer for prosessforbedring basert på erfaringskartleggingen.
- Utvikling av en prosessbeskrivelse for endringshåndtering basert på retningslinjene.
- Utvikling av en verktøyprototype for endringshåndtering basert på prosessbeskrivelsen. Verktøyprototypen støtter kommunikasjonen mellom sluttbrukere, kunde og utviklere på en måte som ingen tilsvarende verktøy for endringshåndtering på det kommersielle markedet gjør.

Dessuten tar verktøyprototypen hensyn til det økonomiske perspektivet rundt endringshåndtering, spesielt på det punkt at kunden involveres i de økonomiske beslutningene rundt hvorvidt et endringsønske skal implementeres eller ikke. Dette skiller også verktøyprototypen fra eksisterende verktøy for endringshåndtering.

Målet med å beskrive erfaringene er at problemer opplevd i prosjektene kan bli unngått av fremtidige, lignende webutviklingsprosjekter. Det kan virke som om anbefalingene er tildels åpenbare, men erfaringer fra de tre prosjektene viste at det kan være nødvendig å understreke viktigheten av slike anbefalinger. Hensikten med utviklingen av prosessbeskrivelsen og verktøyprototypen er å støtte opp under endringshåndtering i evolusjonær webutvikling fordi erfaringsinnhentingene viste at dette var et problemområde.

Mange av resultatene i denne oppgaven er tidligere publisert i (Lien, Arisholm & Skandsen 2001), og ble presentert på konferansen EuroSPI 2001, se forøvrig vedlegg A.

## 1.7 Språkbruk

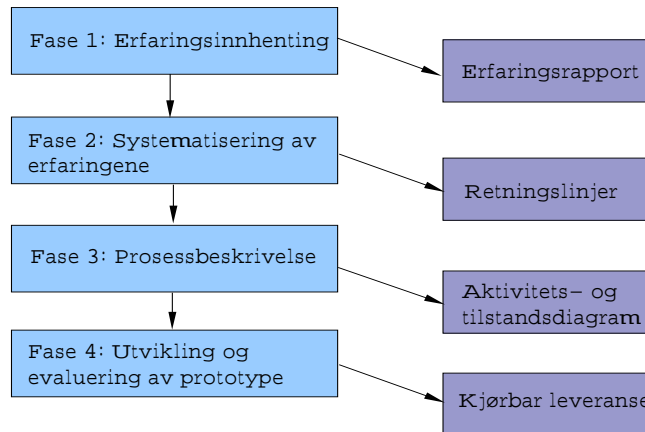
De forskjellige aktørene i et utviklingsprosjekt betegnes gjennomgående i oppgaven som henholdsvis:

- kunde – organisasjonen som betaler for applikasjonen
- brukere – de kommende brukerne av applikasjonen som er under produksjon
- leverandør – organisasjonen som utvikler produktet

Prosjektene er gitt fiktive navn for å respektere deres ønske om å være anonyme. Forslaget til et verktøy som er utviklet i denne oppgaven vil konsekvent betegnes som *verktøyprototypen*.

## 1.8 Metode

Arbeidet bestod av fire faser som ga ulike former for resultater, illustrert i figur 1.2. Fase 1 er det innledende innsamlingsarbeidet der intervjuene ble gjort. De viktigste intervjuene ble transkribert, og det ble laget detaljerte intervjureferater fra alle intervjuene. Utfallet av denne fasen var en erfaringsrapport. I fase 2 ble erfaringene systematisert. Leveransen fra fase 2 bestod av et sett praktiske retningslinjer som fungerer som et innspill til Genova webprosess. I fase 3 ble det utviklet en prosessbeskrivelse for endringshåndtering i evolusjonære prosjekter basert på erfaringsrapporten



Figur 1.2: Metodiske faser i resultatinnsamlingen

og retningslinjene. Resultatet av fase 3 var et aktivitetsdiagram og et tilstandsdiagram for endringshåndtering. I fase 4 ble det utviklet en prototype basert på prosessbeskrivelsen. Resultatet av denne fasen er en kjørbare leveranse. Denne prototypen er ment som et innspill til Genova-verktøyet. Mens utviklingen av prototypen pågikk, ble det foretatt en evaluering av fem testbrukere. Videre utvikling ble utført basert på innspillene.

Det initielle forskningsspørsmålet da innsamlingen av data startet, var å finne ut hvilke problemer man møtte ved en evolusjonær utvikling av webapplikasjoner. Denne hovedoppgaven er en eksplorativ studie der den detaljerte problemstillingen ikke var fastlagt på forhånd (Seaman 1999). Dersom det gjennom innsamlingen av data var emner som utpekte seg som spesielt interessante forskningsmessig, ble det stilt flere spørsmål om dette temaet under intervjuene. Etter den eksplorative utforskningen, viste det seg særlig interessant å fokusere på problemer som oppstod rundt endringshåndtering. Resultatene bærer preg av en eksplorativ tilnæringsmetode. Resultatene er først relativt brede og konsentrerer seg om generelle resultater fra erfaringsinnhenting, før det fokuseres på endringshåndteringen.

Arbeidet startet med intervjuer av:

- utviklere og prosjektledere hos utviklerorganisasjonene
- sluttbrukere
- prosjektledelsen hos kundene

Fokus for retningslinjene ble delvis styrt, samtidig som mulighetene ble holdt åpne for problemer av annen art som hadde dukket opp under webutviklingsprosjektene. Problemområder kom fram ved at de intervjuede fikk prate relativt fritt. Ønsket var å kunne finne retningslinjer til Genova webprosess. Som utgangspunkt for samtalene, ble fokus lagt på brede emner som:

- prosjektetablering og planlegging
- prosessbeskrivelse (roller, aktiviteter, leveranser)
- kravspesifisering
- risikovurdering
- brukermedvirkning

Retningslinjer ble utarbeidet ved å foreta en analyse av erfaringene. Analysen ble gjort ved at de viktigste intervjuene ble transkribert detaljert, samt at detaljert nedskrevne intervjureferater fra alle intervjuene ble utformet. Intervjureferatene ble så analysert og sendt tilbake til respondentene for at de skulle lese gjennom og kontrollere om analysen, etter deres mening, var korrekt. På denne måten ble det kvalitetssjekket om intervjuerens oppfatning var mest mulig i overensstemmelse med det respondenten faktisk ønsket å formidle. Deretter ble de reviderte intervjurapportene sammenlignet i en ny analyse, før de viktigste resultatene i denne analysen ble sammenfattet i en mer overordnet analyse av felles prosjekterfaringer og forslag til retningslinjer. Det er disse mer overordnede prosjekterfaringene som etter en slik analyse er presentert i denne oppgaven.

Gjennom analysen av prosjekterfaringene kom det fram at webutviklingsprosjektene under utviklingen hadde hatt behov for et verktøy som strukturerte endringshåndteringen i prosjektene. Siden det ikke fantes en slik verktøystøtte i Genova-verktøyet fra før, var det et reelt behov i Genova for et slikt verktøy for endringshåndtering. Det ble dermed utarbeidet en UML-modell for hvordan prosessen bak dette verktøyet burde fortone seg, og basert på prosessbeskrivelsen ble en prototype på et slikt endringshåndteringsverktøy implementert. Prototypingen var en iterativ prosess tre testbrukere kom med konstruktive synspunkter på prototypen under utvikling. Prototypen tok form av en kjørbær webapplikasjon. Synspunktene ga viktige justeringspunkter rundt hvordan enkelte elementer ved verktøyet burde være. Når prototypen var ferdigstilt, ble den evaluert av to nye testbrukere som kom med innspill på hvordan den ferdige prototypen fungerte. Prototypen vil i kommende tid bli bearbeidet videre og reimplementert som en del av Genova-verktøyet.

## 1.9 Avgrensning

Denne oppgaven er en del av det subprosjektet innen PROFIT som Genera og Avdeling for Industriell systemutvikling har satt seg fore: å utvikle en Genova webprosess. Da det står utenfor et hovedfags rekkevidde å utvikle en full prosessmodell med tilhørende verktøystøtte, konsentrerer denne oppgaven seg om å gi innspill til generelle retningslinjer for Genova webprosess, og å utvikle en prosessbeskrivelse og en kjørbare verktøyprototype som støtter opp under endringshåndtering i evolusjonær utvikling.

## 1.10 Oppbygning av oppgaven

Resten av oppgaven er organisert som følger:

**Kapittel 2** er et bakgrunnskapittel som beskriver endringshåndtering og evolusjonær utvikling av webapplikasjoner.

**Kapittel 3** omhandler hvilke metoder som er benyttet i denne oppgaven, både når det gjelder metoder for erfaringsinnhenting, bearbeidelse av intervjuer, og utarbeidelse av verktøyprototypen.

**Kapittel 4** er en analyse av erfaringene som ble hentet inn gjennom intervjuer med 15 personer i de tre ulike webutviklingsprosjektene.

**Kapittel 5** beskriver verktøyprototypen som ble utviklet som et løsningsforslag på hvordan man skulle håndtere den ukontrollerte endringshåndteringen som ble beskrevet under erfaringsinnhenting.

**Kapittel 6** konkluderer og oppsummerer resultatene som er kommet fram under oppgaven. I tillegg gis en beskrivelse av videre arbeid som er i gang som en fortsettelse av oppgaven, samt forslag til videre studier.

## Kapittel 2

# Evolusjonær utvikling av webapplikasjoner

Dette er et bakgrunnskapittel som tar for seg teori som finnes i eksisterende litteratur. Først i kapitlet beskrives ulike måter å håndtere endringer på, før det kommer en klassifisering av ulike prosessmodeller som brukes i IT-industrien. Deretter eksemplifiseres de evolusjonære, inkrementelle prosessmodellene gjennom en kort beskrivelse av ulike instanser av dem. Siden Genovaprosessen er en evolusjonær, inkrementell prosessmodell, beskrives ulike instanser av denne type prosessmodell for å gi et sammenligningsgrunnlag og en bredere forståelse av denne type utvikling.

Dette kapitlet fokuserer særlig på hvordan ulike prosessmodeller tar hensyn til at kravene til en applikasjon ofte endrer seg underveis i en utviklingsprosess, samt hvordan de ulike prosessene håndterer endringsforespørsler underveis i utviklingen. Fokus legges på dette fordi skiftende krav og endringshåndtering er tett sammenvevet; nye krav underveis i utviklingen medfører endringsønsker som igjen håndteres på ulike måter i de forskjellige prosessmodeller.

### 2.1 Endringer underveis i systemutviklingen

Endringsforespørsler underveis i utviklingen kan komme både fra kundeorganisasjon, brukere og utviklerorganisasjon. Det kan være mange grunner til at kravene fra kundeorganisasjon eller brukere endrer seg. En av dem er at det kan være vanskelig for kunden og brukerne å kommunisere kravene på en måte som gjør at utviklerne forstår dem (Zamperoni, Gerritsen & Bril 1995). Underveis i utviklingen blir utviklerne så klar over at de har misforstått krav eller arbeidsprosesser. Dermed må kravspesifikasjonen endres. En annen grunn kan være at kunden eller brukerne underveis i utviklingen får økt bevissthet om hvilke behov og ønsker som bør ligge som premiss

for applikasjonen (Ehn 1993). Nye krav kan også dukke opp under systemutviklingsprosessen fordi kunden eller brukerne kommer over en ny tjeneste med en funksjonalitet de ikke var klar over, og dermed ønsker seg noe tilsvarende (Cloyd 2000, Grudin 1991). IT-bransjen er i et dynamisk marked der mulighetene for bedre funksjonalitet og mer innbydende brukergrensesnitt gror i et høyt tempo – noe som kan føre til økte forventninger fra brukerne (Grudin 1991). Dette gjelder spesielt webapplikasjoner, fordi man her opererer i en skiftende verden der nye tjenester stadig dukker opp, samtidig som ulike brukergrupper har lettere tilgang til tjenester enn det som er tilfelle ved tradisjonelle applikasjoner (Ward & Kroll 2000). Den enkle tilgangen gjør også at kunden og brukerne har større kunnskaper om hvilke muligheter en webapplikasjon har.

Det er ikke bare endringsforespørsler fra kunden eller brukerne som kan medføre endringer på produktet. Endringer kan også oppstå internt i utviklerorganisasjonen som en følge av flere faktorer, blant andre:

- endrede lover og reguleringer
- testing som avslører produktfeil
- integrasjonsproblemer ved ulike systemkomponenter
- tilpasning til ny hardware, nye operativsystemer eller grensesnitt mot andre systemer
- teknologien fungerer annerledes enn først antatt
- mangelfull teknologi medfører endringer i design eller arkitektur
- konkurrerende produkter medfører taktiske endringer

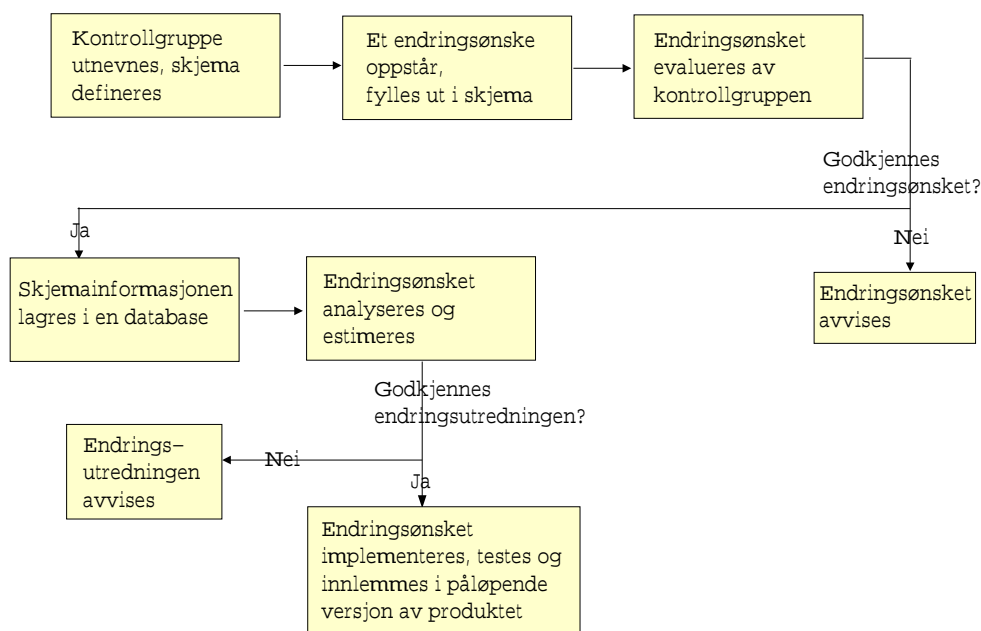
(Kruchten 2000, Lientz et al. 1978, Arisholm 2001).

## 2.2 Formalisering av endringshåndteringen

Endringshåndtering ble i introduksjonskapitlet beskrevet som en aktivitet for å kontrollere og holde oversikt over forespørsler om å endre en produktdel eller en produktprosess. Et endringsønske har en opphavsmann, samt en årsak. Ofte har et endringsønske også en alvorlighetsgrad og en prioritet. Livsløpet til et endringsønske består av ulike tilstander, for eksempel *nytt*, *loggført*, *godkjent*, *tilordnet* og *ferdigstilt* (Kruchten 2000). Mens endringsønsket gjennomgår ulike tilstander, kan det oppstå ny informasjon om endringsønsket (Kruchten 2000). Det kan derfor være hensiktsmessig å lagre informasjon om endringsønsker i en database, slik at man på ulike stadier i utviklingen kan hente ut informasjon om et spesifikt endringsønske. På denne måten kan man også lage statusrapporter.



En prosess rundt endringshåndtering kan formaliseres på ulike måter. En formalisert endringsprosess innebærer at alle endringsønsker evalueres med hensyn på behovet for endring, virkningen endringsønsket vil ha på prosjektet og på systemet, hvilke tidsrammer endringen kan utføres innen med mer (Sjøberg 1993). En standardisert prosess for endringshåndtering i et prosjekt kan sørge for at endringer blir gjort på en konsistent måte, og at de involverte parter informeres om produktendringer, kostnader og estimatrevideringer som er en følge av ulike endringsforespørsler (Kruchten 2000). Nedenfor gjennomgås to ulike forslag på hvordan en prosess rundt endringshåndtering kan være. Forslag til prosesser for endringshåndtering som finnes i litteraturen er ganske like. Av den grunn er bare disse to forslagene tatt med.



Figur 2.1: Sommervilles forslag til en prosess for endringshåndtering

Sommerville foreslår følgende prosess for endringshåndtering (Sommerville 1995), illustrert i figur 2.1: først utnevnes det en kontrollgruppe som er ansvarlig for endringshåndteringen. Kontrollgruppen kan bestå av en prosjektleder og et par utviklere. Denne gruppen definerer et skjema for å fylle inn endringsønsker. Første stadium i Sommervilles prosess for endringshåndtering er at dette skjemaet for endringsforespørsler fylles ut av den som kommer med endringsønsket. Dette skjemaet foreslår han kan inneholde felter om hvem som sendte inn ønsket, dato, en endringsbeskrivelse, en endringsvurdering, en endringsprioritet og et foreløpig endringsestimat. Skjemaets gyldig-

hetsgrad evalueres så av kontrollgruppen som håndterer endringsforespørsler. Årsaker til at skjemaet kan være ugyldig, er for eksempel at endringen vedrører en produkt del som ikke lenger inkluderes i neste versjon av produktet, at endringsforespørselen skyldes en misforståelse snarere enn en systemfeil, at endringen allerede er kjent, eller at endringen allerede er implementert og inkludert i neste leveranse.

Neste steg i Sommervilles forslag er å lagre skjemainformasjonen i en database. Deretter foreslår Sommerville at det lages en analyse av endringen med hensyn på innvirkningen den vil ha på resten av systemet, samt en estimatvurdering. Sommerville foreslår at dette arbeidet sendes til kontrollgruppen for endringshåndtering som bestemmer hvorvidt endringen aksepteres eller ikke. Dersom endringsønsket aksepteres, sendes det til et utviklings- eller vedlikeholdsteam for implementasjon. Endringsønsket implementeres, testes og innlemmes i påløpende versjon av produktet. Sommerville understreker at denne prosessen er svært rutinepreget, og derfor er egnet for å understøttes av et verktøy for endringshåndtering.

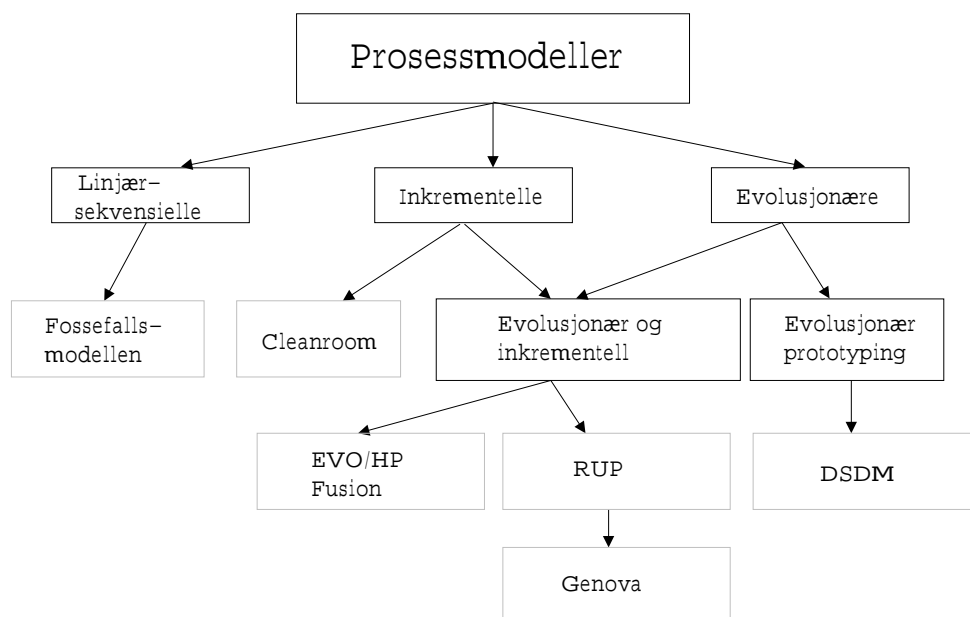
Kruchten foreslår en lignende prosess for endringshåndtering (Kruchten 2000). Han foreslår at man først utnevner en kontrollgruppe for endringshåndtering som består av ulike representanter fra de involverte parter — inklusive kunder, utviklere og brukere. Denne gruppen har som oppgave å vurdere hvert enkelt endringsønske. Gruppen godkjenner eller avviser de ulike endringsønskene som kommer inn. Ved godkjenning av endringsønsket tilordner kontrollgruppen det en prioritet og en ansvarlig utvikler. Utvikleren overtar så endringsønsket, og er ansvarlig for implementasjon, test og innlemming av det nye endringsønsket i en ny leveranse av systemet. Et endringsønske er ferdigstilt når endringen har blitt utført, testet og inkludert i en leveranse (Kruchten 2000). Kruchten mener denne prosessen egner seg godt for verktøystøtte.

Forskjellen mellom Sommerville og Kruchtens forslag til prosesser for endringshåndtering, er hovedsaklig at Kruchten innlemmer kundeorganisasjon og brukere i endringshåndteringsprosessen. Sommervilles prosess egner seg best for endringshåndtering internt i utviklerorganisasjonen, både fordi skjemaet han foreslår inneholder felter som det for en bruker ville være vanskelig eller uansvarlig å fylle ut (feltene for endringsprioritet og et foreløpig endringsestimat), samt at hans forslag ekskluderer kundeorganisasjonen fra å delta i bestemmelsen om hvorvidt et endringsønske skal implementeres eller ikke. Kruchtens endringshåndtering tillatter at både brukere, kunde og utviklerorganisasjon kan sende inn endringsønsker, og tillatter de samme parter å være med på å bestemme hvorvidt et endringsforslag skal godkjennes eller ikke.

## 2.3 Prosessmodeller

En prosessmodell kan betegnes som en generell beskrivelse av hvilke prosjektaktiviteter som bør være med i utvikling av programvare, og i hvilken rekkefølge prosjektaktivitetene bør utføres. Boehm definerer en prosessmodell som en beskrivelse av transaksjonskriteriene for når man bør gå fra en fase til den neste under programvareutvikling (Boehm 1988). Det finnes ikke noe slikt som en riktig eller gal prosessmodell, men noen prosesser kan være mer passende for utviklingen av et bestemt type produkt enn andre.

Dersom man ikke følger en bestemt prosessmodell, består utviklingsprosessen av å skrive litt kode, og deretter fikse problemene som oppstår i koden. Denne måten å utvikle systemer på følger ingen bestemt prosessmodell, og dette kan ha flere konsekvenser (Boehm 1988). For det første blir koden som skrives gjerne svært ustrukturert, Boehm kaller det treffende for “spagettikode”. For det andre mangler denne måten å utvikle på en analyse av problemet som skal løses, noe som lett medfører at man ikke klarer å definere brukerens *egentlige* krav (Boehm, 1986).



Figur 2.2: Kategorisering av ulike prosessmodeller

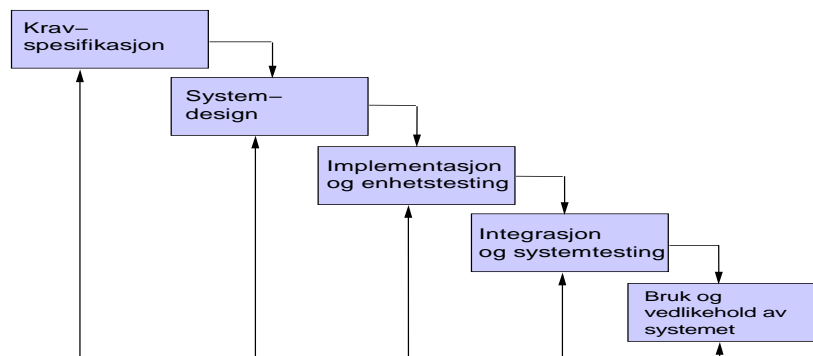
Arisholm har gjort en inndeling i forskjellige prosessmodeller som er noe modifisert i denne oppgaven (Arisholm 2001). Inndelingen vises på figur 2.2. Denne figuren kategoriserer vanlige prosessmodeller. Arisholm deler prosess-

modellene inn i kategoriene lineærsekvensielle modeller, inkrementelle modeller og evolusjonære modeller. Pilene viser hvordan en prosessmodell arver egenskaper fra en prosessmodell høyere opp i hierarkiet. Under beskrives disse kategoriene. Det er særlig lagt vekt på evolusjonære prosessmodeller, da disse er fundamentet for oppgaven. Det fokuseres også på hvordan de forskjellige prosessmodellene tar hensyn til brukerens egentlige behov, hvordan de prøver å tilpasse seg at brukernes krav ofte skifter underveis i utviklingen, og hvordan dette medfører økt behov for endringshåndtering.

### 2.3.1 Lineærsekvensielle modeller

#### Fossefallsmodellen

Den mest brukte av lineærsekvensielle modeller er fossefallsmodellen (Royce 1970). Fossefallsmodellen deler systemutviklingen inn i separate, klart atskilte faser, som vist på figur 2.3, hentet fra Sommerville (Sommerville 1995). Avslutningen av en fase markeres gjerne med en milepæl og fyldig dokumentasjon. Kunden og eventuelt sluttbrukere er med på å skrive kravspesifikasjon helt i starten av utviklingen, og kommer inn mot slutten av utviklingen for å delta i testing. Fossefallsmodellen er hyppig brukt i industrien. Dette kan skyldes at dette er en svært synlig prosess, slik at det er lett for en kunde å forholde seg til den.



Figur 2.3: Fossefallsmodellen

Boehm argumenterer for at problemet med fossefallsmodellen er den statiske inndelingen av prosjektet i atskilte faser, og at brukeren ikke er med i utviklingen underveis i prosjektet (Boehm 1988). Den skarpe faseinndelingen kan gjøre det vanskelig å oppdage underveis at brukerens krav egentlig er annerledes. Siden man ikke lager deler av produktet ferdig, men arbeider med hele produktet stegvis fase for fase, har man ikke anledning til å vise fram deler av produktet for kunden for å få feedback på om man har forstått kravene

på riktig måte. Brukeren kommer inn for å evaluere prosjektet veldig sent i utviklingen, og hvis man da har misforstått kravene, kan det være fatalt. Det er da svært kostbart å gå tilbake og rette opp grunnleggende arkitektur eller design, fordi prosjektet er kommet så langt i utviklingen. Man risikerer dermed at produktet ikke møter kundens virkelige krav.

I tråd med Boehm, mener Pfleeger at det største problemet med fossefallsmodellen er at den ikke reflekterer hvordan systemutvikling egentlig foregår (Pfleeger 1998). Vanligvis er systemutvikling en *problemløsende* prosess – en løsning på et problem som aldri har vært løst før. En problemløsende tilnærming krever ifølge Pfleeger en god del “fram-og-tilbake”-aktiviteter (iterasjon) fordi man oppdager nye sider ved problemet som man ikke har tenkt på før. Da må man gjerne gå tilbake og gjøre endringer på tidligere avgjørelser. Her spiller brukerne en sentral rolle. Fossefallsmodellen svikter på dette feltet, fordi dens statiske inndeling i faser ikke gir noen støtte for at man kan gå tilbake til en tidligere fase og forandre på deler av produktet etter innspill fra brukeren. Dermed blir det vanskelig å motta innspill fra brukere underveis i prosessen og forandre aktiviteter eller kravspesifikasjon i henhold til nye krav og ønsker som har dukket opp underveis. Fossefallsmodellen åpner med andre ord ikke for den interaksjonen med brukeren man kan ønske seg dersom kravene er vage og usikre. Fossefallsmodellen gir ingen veiledning for “fram-og-tilbake”-aktiviteter – problemløsende aktiviteter som kan gjøre at brukeren i større grad får et produkt tilpasset sine behov.

Zamperoni et. al mener kommunikasjon mellom utviklere og brukere er tett sammenknyttet med problemet om hvordan man nedskriver en riktigst mulig kravspesifikasjon (Zamperoni et al. 1995). Av og til har brukerne en helt spesiell visjon av produktet de ønsker seg, men klarer ikke å formulere dette. Dette problemet møtes ikke på en særlig hensiktsmessig måte i fossefallsmodellen, der brukerne kun er med i en kort fase i begynnelsen og i slutten av produktutviklingen. Den sene tilbakemeldingen på systemet er en stor risiko å løpe. Hvis man da ikke har møtt brukernes ønsker på en tilfredstillende måte, kan det bli svært vanskelig og dyrt og gå tilbake i utviklingen for og rette på dette.

Et annet problem med denne modellen, er understrekingen av at hver fase bør avsluttes med fyldig dokumentasjon, og at denne dokumentasjonen sees på som et tegn på at fasen er avsluttet (Sommerville 1995). Denne dokumenteringen gjør det vanskelig å gå tilbake til utviklingen i en tidligere fase. Dermed blir det også vanskelig å integrere nye, skiftende krav fra brukeren. Den dokumentbaserte standarden kan også tvinge utviklere til å skrive spesifikasjoner på krav som er lite utforsket, noe som dermed kan medføre at det produseres store mengder med ubrukelig kode.

Man kan imidlertid ikke neglisjere innflytelsen de tradisjonelle lineærsekvensielle prosessmodellene har hatt på systemutvikling. For ledelsen gir de god

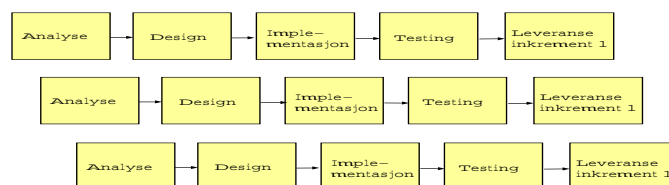
kontroll over utviklingen, slik at det er mulig å overvåke hvorvidt man holder seg til milepælsplanen (Zamperoni et al. 1995).

### 2.3.2 Mindre rigide prosessmodeller

For å prøve å bøte på problemet med at brukerne ikke alltid er i stand til å formulere hva de ønsker seg, har flere prosessmodeller vært foreslått, blant andre inkrementell prosessmodell, evolusjonær prosessmodell og en blanding av disse. Her prøver man å ha en kontrollert evolusjon av systemet, men med respons fra brukerne underveis i hele prosessen. Brukerne skal altså ikke bare delta på begynnelsen og ved slutten av utviklingen, slik som under fossefallsmodellen. Felles for disse modellene er at produktet deles opp i mindre biter, og at disse delene utvikles enten separat eller i parallell. Disse prosessmodellene atskiller seg fra fossefallsmodellen gjennom å åpne for muligheten for parallell utvikling av moduler, parallell gjennomføring av ulike faser, og tilbakemelding på deler av produktet fra brukeren underveis i utviklingen. I motsetning til fossefallsmodellen, gjør man seg dermed ikke ferdig med all aktivitet som hører til en fase før man starter med neste fase.

#### Inkrementell utvikling

Følger man en inkrementell prosessmodell, lager man i begynnelsen av systemutviklingen en systemarkitektur, og baserer kravspesifikasjon og utvikling av inkremerter på denne. For hvert inkrement utfører man så aktivitetene analyse, design, koding og testing. Figur 2.4, hentet fra Arisholm, skisserer denne utviklingsprosessen grafisk (Arisholm 2001).



Figur 2.4: Inkrementell utvikling

Man går ikke tilbake på inkrementet og forandrer det, men bruker tilbakemeldinger som innspill og korreksjon for de neste modulene som skal utvikles. Med en gang inkrementet er ferdigutviklet, blir grensesnittet dens frosset, slik at det kan benyttes av de kommende inkrementene. Har man levert og testet et inkrement, forandres ikke dette med mindre man oppdager feil.

En effektiv måte å utnytte potensialet i den inkrementelle prosessmodellen på, er å levere inkremerter bestående av de mest kritiske delene ved applikasjonen først. Dette kan enten være inkremerter som er basert på vage

brukerkrav, eller inkremer som tester ut tekniske aspekter ved produktet man skal utvikle. Når man så har levert en del til brukeren, kan brukeren eksperimentere med modulen og gi feedback til utviklerne, slik at utviklerne har dette i minne for de inkrementene de skal utvikle videre (Sommerville 1995).

Fordi man ikke forandrer på et utviklet inkrement, unngår den inkrementelle prosessmodellen problemer med en kontinuerlig forandring av inkrementenes grensesnitt. Sommerville mener det er lett å ha prosjektstyring over denne modellen, fordi planer og dokumenter produseres for hvert inkrement. Prosessen åpner også for tilbakemeldinger fra brukerne tidlig i systemutviklingen.

Et problem med den inkrementelle utviklingsmodellen, er at systemarkitekturen må etableres før alle kravene er spesifisert, og før man har fått noe feedback på produktet fra kunden/brukerne. Dette betyr at kravspesifikasjonen og utvikling av inkremer begrenses av arkitekturen, som så kan være basert på en ikke fullstendig avklart kravspesifikasjon. Fordelen med modellen er at den kan være en effektiv måte å redusere leveringstiden på og få produktet ut på markedet fordi man kjører en parallell utvikling (Arisholm 2001). Et eksempel på en inkrementell prosessmodell er "Clean-room"-prosessen (Linger 1993). Den foregår som beskrevet overfor ved at man fryser et inkrement etter levering, og baserer grensesnittet til kommende inkremer på grensesnittet som er ferdigutviklet.

### **Evolusjonær utvikling**

Evolusjonær utvikling går ut på å implementere og levere en kjørbare leveranse til bruker så tidlig som mulig og deretter måle nytten brukeren har av det foreløpige systemet (Thomassen 1996). Basert på brukerens respons og endringsønsker, justeres så både design og formål, før leveransen videreutvikles og igjen leveres til brukeren for en ny evaluering. Evolusjonær utvikling ser dermed på programvareutvikling som en syklisk sekvens der man i hver ny runde i syklusen (re)designer, (re)implementerer og (re)evaluerer (Floyd 1984). I motsetning til en inkrementell utvikling, ser man altså at det her utvikles videre på leveranser som er levert til brukeren. Som i inkrementell utvikling konstrueres man først modulene med høyest risiko forbundet til seg.

Man understreker i evolusjonær utvikling at systemet lages i et dynamisk og skiftende miljø. I stedet for å ta høyde for alle krav på forhånd, bygger man derfor systemet så det kan inkludere nye og uventede krav. Antallet syklener man trenger i evolusjonær utvikling situasjonsbestemmes (Floyd 1984). Ifølge Floyd eksisterer det derfor ingen vedlikeholdsfase i evolusjonær systemutvikling. I stedet er denne fasen erstattet av videre utviklingssykluser basert på eksisterende systemversjoner og nye krav.

En evolusjonær utviklingsmetode er basert på at man tidligst mulig skal få signaler om at produktet ikke er i tråd med brukernes behov, slik at det er

mulig å justere utviklingen på et så tidlig stadium som mulig. Et av de viktigste kriteriene for å kunne benytte seg av evolusjonær utvikling er at man bygger opp en åpen systemarkitektur (Thomassen 1996). Dette innebærer at man til enhver tid kan bygge ut, rette opp eller forandre systemet uten at det fører til for mye merarbeid. Løkken med å levere inkrementet til brukeren for respons fortsetter helt til brukeren er fornøyd med produktet — eller mer realistisk: til kunden ikke har mer penger å gi (Sommerville 1995).

Boehm kritiserer evolusjonær utvikling for å produsere “spagettikode” og urealistiske krav fra kunden (Boehm 1988). Studier har vist at evolusjonær utvikling passer best for relativt små systemer, eller som en prosessmodell for delutvikling av et større system (Sommerville 1995).

### **Evolusjonær prototyping**

Evolusjonær prototyping kan klassifiseres som en underkategori av evolusjonær utvikling (Arisholm 2001). Evolusjonær prototyping kan sees på som utvikling i “versjoner” (Floyd 1984), der prototyper videreutvikles og blir det endelige systemet (Arisholm 2001, Lichter, Schneider-Hufschmidt & Zullighover 1994). I evolusjonær utvikling legges det vekt på at systemet bygges gjennom evolusjon (Floyd 1984). Vekten legges på at man gradvis utvikler et system som underveis tilpasser seg skiftende krav — krav som ikke kan bestemmes i en tidlig fase. Evolusjonær prototyping er fundamentert på at organisasjoner forandrer seg, og at nye krav derfor kan oppstå, samtidig som en interaktiv applikasjon, når den er tatt i bruk, omformer konteksten den brukes i og kan gi opphav til nye krav (Floyd 1984). Floyd mener evolusjonær prototyping bryter med en lineær ordning av utviklingstrinn og gjør utviklingen mer syklisk. Antallet sykler man benytter kan variere fra prosjekt til prosjekt, avhengig av hva det er behov for.

Mathiassen og Forsth definerer en prototype slik:

*“En prototype er en funksjonerende modell av et framtidig system, som skal kunne bearbeide reelle data korrekt.”*

(Mathiassen & Forsth 1988)

Selv om en prototype ikke er et komplett system, skal den likevel for brukeren fungere som det. Når en prototype brukes, skal brukeren og systemutviklerne sammen kunne se på, vurdere og forbedre prototypen. Formålet med å eksperimentere med prototyper er å skape erfaringer og konstruktivt anvende disse erfaringene i det videre arbeide. Lindgaard skiller mellom statiske og dynamiske prototyper — også kalt horisontale og vertikale prototyper (Lindgaard 1994, Bødker, Kensing & Simonsen 2000). Statiske prototyper inneholder ingen funksjonalitet, og er enkle å lage og å endre. En statisk prototype kan realiseres som skjermbilder på en overhead, eller som “flate”



skjermbilder ved hjelp av et presentasjonsprogram eller et grafisk program. Statistiske prototyper er kun ment for demonstrasjon og mangler funksjonalitet som virker tilfredsstillende. Dynamiske prototyper tillatter derimot at brukeren interagerer med systemet. En slik prototype er et enkelt, men kjørende IT-system som illustrerer og simulerer utvalgte deler av et kommende system. En dynamisk prototype gjør det mulig å arbeide med virkelige data og å anvende prototypen i egentlige arbeidssituasjoner. Det er derfor lettere å trekke konklusjoner om det kommende systemet ved dynamisk prototyping enn ved statisk prototyping. Prototypen som ble utviklet for denne oppgaven er dynamisk. Den tillatter at brukeren gjør operasjoner mot en database, og har implementert all nødvendig funksjonalitet. En fordel ved å bruke dynamiske prototyper kontra statiske er at prototypen på utvalgte områder viser konkrete og håndgripelige egenskaper som ligger tett inntil egenskaper som det ferdige systemet vil ha (Bødker et al. 2000).

### **Eksplorativ, eksperimenterende og evolusjonær prototyping**

Innen dynamiske prototyper skiller Floyd mellom *eksplorativ*, *eksperimenterende* og *evolusjonær* prototyping (Floyd 1984).

- Eksplorative prototyper fokuserer på de grunnleggende problemer rundt kommunikasjon som oppstår mellom utviklere og brukere i de tidligste stadiene av programvareutvikling. Utviklerne har her liten kunnskap om applikasjonsdomenet, mens brukerne ikke har noen klar ide om hva datamaskinene kan gjøre for dem. I denne situasjonen kan en praktisk demonstrasjon av mulige systemfunksjoner være en katalysator for å fremme gode ideer og for å skape grobunn for et kreativt samarbeid mellom de ulike partene som er involvert. En eksplorativ prototype oppstår gradvis som et resultat av en kommunikasjonsprosess der utviklerne demonstrerer mulig funksjonalitet og brukerne tester ut denne funksjonaliteten med tanke på deres arbeidsprosesser. En eksplorativ prototype er ifølge Floyd ofte rotete og ustrukturert, og vil derfor normalt kastes. Siden en eksplorativ prototyping ikke har noen relasjon til det ferdige produktet, trenger ikke verktøyene man bruker for å utvikle prototypen være de samme som man bruker for å utvikle det endelige systemet. Eksplorativ prototyping er også kalt *throw-away prototyping* (Sommerville 1995).
- I eksperimenterende prototyper legges vekten på å bestemme hvorvidt en foreslått løsning er adekvat før man investerer i en oppskalert implementasjon av systemet. En løsning på et gitt problem foreslås gjennom en prototype, og prototypen evalueres så gjennom eksperimenterende bruk. En måte å lage en eksperimenterende prototype på er ved hjelp av hel eller delvis funksjonell simulering. I en slik prototype er det mest eller all funksjonalitet implementert, men man har ikke tatt hensyn

til effektivitetskrav eller ytelseskrav. Dermed vil prototypen være uegnet som et endelig system. Den er derimot god for å teste en hypotese om et system.

- I evolusjonær prototyping legges vekten på å la prototypen utvikles i versjoner til et ferdig system ut i fra skiftende krav som ikke kan bestemmes i en tidlig fase av utviklingen. Denne underkategorien av prototyping grenser mot å være en egen prosessmodell, og ble derfor beskrevet i kapittel 2.

I følge Floyds kategoriseringer er verktøyprototypen utviklet i denne oppgaven et eksempel på en eksperimenterende prototype. Prototypen er foreslått som en løsning på et identifisert problem – ustrukturert endringshåndtering i evolusjonære utviklingsprosjekter. Metoden som ble brukt var i utgangspunktet full funksjonell simulering, men etter eksperimenterende bruk med prototypen er det oppdaget ny funksjonalitet som burde inkluderes i en endelig versjon av prototypen. I prototypen er det heller ikke vektlagt noen ikke-funksjonelle krav, og kildekoden som er implementert, er ikke egnet til bruk i et kommende verktøy for endringshåndtering.

### **Fordeler og ulemper ved prototyping**

Prototyping demonstrerer en operasjonell versjon av et kommende system for at både utviklere og brukere kan vurdere systemet, komme med endringsforslag og evaluere hvorvidt systemet bør implementeres eller ikke (Floyd 1984). Prototyper introduserer dermed muligheten for kommunikasjon og feedback (Floyd 1984, Jørgensen 1984). Utviklere kan få kommunisert sine ideer til brukere, mens brukere kan teste ut ideer for å se om systemet er noe de kan tenke seg å bruke. Ved prototyping må også utviklere forklare sine valg for brukere som kan stille spørsmålsteget ved dem. Dette gjør at valgene kanskje blir omgjort dersom man innser at de var feil (Jørgensen 1984). Utviklere kan dermed finne fram til manglende eller inkonsistente krav etter som prototypen utvikles (Sommerville 1995) Prototyping gjør det mulig å ta med tilbakemeldinger fra brukerne tidlig i designprosessen, og å evaluere flere iterasjoner av et design (Dumas & Redish 1999, Floyd 1984).

Systemutvikling er en prosess der man lærer. Gjennom utvikling av et produkt får man ny innsikt og forstår hvordan man kunne gjort deler av produktet bedre (Floyd 1984). Gjennom prototyping får man sjansen til å lære av sine feil før man starter en full implementasjon av et system (Jørgensen 1984), og man kan ta læringen med seg inn i utviklingen. Funksjonalitet som er vanskelig å bruke eller forvirrende kan ved prototyping bli identifisert og forbedret i en full versjon av systemet (Sommerville 1995).

Negative aspekter ved prototyping er at manglende funksjonalitet kan gjøre at en bruker ikke forstår kompleksiteten til det kommende systemet, og slik

sett tror det vil bli enklere å bruke enn det i virkeligheten vil (Jørgensen 1984, Bratteteig 1997). Det kan også være vanskelig for den som har utviklet prototypen å kaste det den har gjort, og innse feilene som ble gjort under utviklingen. En prototype skal kunne forandres, ikke frosses, og det kan være vanskelig for den som har utviklet prototypen å akseptere dette (Jørgensen 1984).

## **Evolusjonære, inkrementelle utviklingsprosesser**

Evolusjonær, inkrementell utvikling kan klassifiseres som en hybrid av inkrementelle og evolusjonære prosessmodeller, der elementer fra begge de to metodene kombineres (Arisholm 2001). Dette kan innebære at man ikke nødvendigvis fryser et inkrement, men kan redesigne, re-implementere og re-evaluere hvert inkrement mens man lar produktet gradvis utvikles mot et kjørbart system. Fellestrekket for evolusjonære, inkrementelle utviklingsprosesser er at de har som mål å identifisere de *virkelige* kravene til brukerne ettersom systemet utvikles. Misforståelser og vage krav forsøkes avdekkes så tidlig som mulig ved hjelp av å utvikle inkremitter av produktet som leveres til kunden (eventuelt interne medarbeidere) for evaluering. Hvis kunden ikke er fornøyd, kan man begynne en ny iterasjon der man endrer funksjonalitet for bedre å tilpasse dem til kundens krav, og eventuelt lage tillegg-funksjonalitet. På denne måten håper man å identifisere brukernes egentlige krav ettersom utviklingen pågår. For hver iterasjon skal kunden kunne se en ny versjon av programvaren. Dette gjøres for å redusere risikoen forbundet med systemutvikling. Eksempler på slike utviklingsprosesser er Boehms spiralmodell, Gilbs EVO (Gilb 1988), HP Evolutionary Fusion (Cotton 1996) og Rational Unified Process (RUP) (Kruchten 2000) og Genova (Arisholm, Skandsen, Sagli & Sjøberg 1999) som er en nedskalert versjon av RUP.

Boehms spiralmodell var den første inkrementelle, evolusjonære utviklingsmodellen (Boehm 1988). Spiralmodellen var et forslag til hvordan man skulle identifisere risiki på et så tidlig stadium i prosjektutviklingen som mulig i utviklingen. Evolusjonær, inkrementell utvikling er siden blitt en modell for å håndtere risiko i forbindelse med for eksempel innføring av ny teknologi, oppfølging av prosjektframdrift, evaluering av design/arkitektur og uklare eller endrede krav (Sjøberg & Arisholm 2000).

## **2.4 Ulike evolusjonære, inkrementelle prosessmodeller**

### **2.4.1 Boehms spiralmodell**

Boehms spiralmodell ble foreslått som en modell som eksplisitt skulle ta høyde for risiko på prosjekter. Det er dette hensynet til risiki som skiller

denne modellen mest fra andre prosessmodeller. Modellen er en prosess som går i spiral, hvor hver runde i spiralen gjennomgår fire faser. I første fase legges mål og planer for runden, og risiko og implementasjonsmuligheter identifiseres. Etter dette lager man en detaljert risikoanalyse, før man lager en utviklingsmodell som tar hensyn til de identifiserte risikoene. I fjerde fase evalueres produktet og prosjektet så langt som det er kommet, og man vurderer hvorvidt man bør starte med en ny runde i spiralen (Boehm 1988).

Modellen er en generell modell, en slags metamodell som kan ligge som fundament for andre modeller, for eksempel som fundament for EVO, HP Fusion, Rational Unified Process (Arisholm 2001), eller som fundament for en blanding av flere av disse (Boehm 1988). Hvis det i et prosjekt for eksempel er stor sannsynlighet for at kundens krav er godt forstått, ved at man for eksempel bare skal transformere en tradisjonell applikasjon om til en webapplikasjon, kan man anvende fossefallsmodellen på toppen av spiralmodellen. En god ting ved spiralmodellen er at den fokuserer på alternative løsninger tidlig i prosessen (Boehm 1988). En ulempe med den er derimot at det er vanskelig å lage en kontrakt på den, fordi den krever fleksibilitet og frihet.

#### 2.4.2 EVO og HP Fusion

EVO vektlegger små og hyppige leveranser, og utvikling i inkremerter (May & Zimmer 1996) som skal leveres til kunden for feedback. Hvert inkrement skal utgjøre mindre enn fem prosent av prosjektutviklingen, men skal likevel inneholde alle de tradisjonelle systemutviklingsaktivitetene, det vil si analyse, design, koding og testing. De *enkleste* delene av applikasjonen utvikles først.

HP Fusion bygger på EVO, men slakker litt av på kravene om brukerrespons og leveranser. Utviklingen av et inkrement skal vare 1-4 uker før det leveres for feedback. I stedet for å vise ethvert inkrement til brukeren, som i EVO, åpner modellen for at en del av inkrementene evalueres av andre medarbeidere i organisasjonen (Cotton 1996). Tilbakemeldingen på inkrementet brukes som planlegging for utviklingen av neste inkrement. Man utvikler også de mest kritiske delene av applikasjonen først, i motsetning til EVO.

Fordelen med EVO/HP Fusion er at den intense tilbakemeldingen gir gode muligheter til å oppdage svakheter ved kravspesifikasjonen, og justere utviklingen etter tilbakemeldingen fra brukerne. Sannsynligheten for å møte brukerens egentlige krav er dermed relativt høy. Ulempen er at modellene krever at ledelsen er villig til å bruke så mye som en tredjedel av deres tid på å motta respons på inkrementene (May & Zimmer 1996), noe som er en markant kostnad.

### 2.4.3 Rational Unified Process(RUP)

The Rational Unified Process (RUP) er en evolusjonær og inkrementell prosessmodell, utviklet av Rational Software (Kruchten 2000). RUP har sitt fundament i spiralmodellen, og ser på systemutvikling som en serie av iterasjoner der utviklingen skjer i inkremitter. Hver iterasjon skal ende opp i en kjørbare leveranse, som er støttet opp av brukermanualer og dokumentasjon. Hver iterasjon består av planlegging, analyse, design, implementasjon og varierende grad av testing, avhengig av hvor langt i prosjektutviklingen man er kommet. Hvis ønskelig, utvikler man inkrementene i parallell. Målet til RUP er å lage programvare som møter behovene til sluttbrukerne innen forutsigbare tids- og budsjettammer (Kruchten 2000).

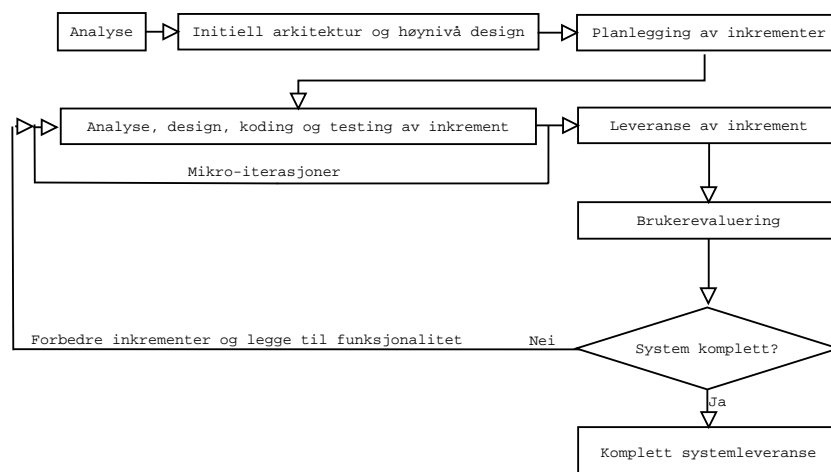
RUP har også utviklet en egen versjon av prosessmodellen som skal gi støtte for webutviklingsprosjekter (Ward & Kroll 2000). Her legges det særlig vekt på at en webapplikasjon bør være visuelt innbydende og konsistent. Grunnen til dette er at det er viktig å fange en brukers oppmerksomhet og interesse umiddelbart når brukeren kommer inn på siden, fordi brukere av WWW er ustabile brukere som lett kan surfe av gårde til en annen, mer innbydende website. Når det gjelder brukermedvirkning på webutviklingsprosjekter, vektlegger RUP at det vil være mange ulike brukergrupper som vil komme til å bruke applikasjonen. Dermed er det viktig å prøve å ta høyde for flere ulike grupper, og få kontakt og dialog mellom flere ulike typer av brukere. Dette står i kontrast til utvikling av vanlige applikasjoner, der det som regel er en mer homogen brukergruppe. RUP foreslår å utvikle flere ulike visuelle løsninger av siden og vise dette i en prototypeaktig maner for brukerne, for å sikre en best mulig visuelt innbydende applikasjon.

RUP er et *prosessrammeverk* som kan adopteres og utvides eller forminskes, modifiseres, minimeres eller ekspanderes for å tilpasse den de spesifikke behovene, karakteristikaene, begrensningene og mulighetene i den enkelte organisasjon (Kruchten 2000). Det er en slik prosessjustering Genera har gjort i sin utvikling av Genovaprosessen.

### 2.4.4 Genovaprosessen

Initielt var Genovaprosessen definert som en nedskalert versjon av RUP. Antagelsen bak dette var at en slik "lettvekterprosess" for evolusjonær utvikling ville øke sannsynligheten for suksess i mindre utviklingsprosjekter (Arisholm, Skandsen, Sagli & Sjøberg 1999). Tanken bak Genovaprosessen er at den skal være en modell for mindre utviklingsprosesser. Dette er basert på en oppfatning i bedriften Genera at antall leveranser, roller og aktiviteter i RUP er for mange til at de er av noen praktisk verdi for små utviklingsprosjekter. Grunnen til å basere den på RUP var også delvis motivert av den økende populariteten denne prosessen har i industrien. Begrepene i RUP begynner å bli en "DeFacto"-standard, og det er lettere å få gjennomslag for en prosess

som bruker terminologi som allerede er godt kjent blant utviklere i industrien. Et annet ønske var å definere en modell som var enklere å sette seg inn i en RUP. Figur 2.4 på neste side illustrerer Genovaprosessen.



Figur 2.5: Flyttdiagram over Genovaprosessen

Prosessen starter med at man lager en initiell arkitektur og et høynivå design. Hvert inkrement utvikles så gjennom iterasjon av alle hovedaktiviteter i en utviklingsprosess — med andre ord analyse, design, koding og testing. Hvert inkrement kan ha opptil tre mikroiterasjoner før det sendes til kunden for validering. Etter å ha gjort en analyse, grovskisse av design, planlagt arkitektur og lagd planer for inkrementene, begynner man på utviklingen av inkrement 1. I inkrement 1 gjør man analyse, design, kode og test på deler av funksjonaliteten av produktet eller eventuelt hele produktet dersom det er rom for det. Etter å ha levert inkrementet til kunden (eventuelt sine medarbeidere) for evaluering, mottar man forslag fra kunden om hvordan produktet bør forbedres og starter disse utvidelsene og forbedringene av produktet på neste inkrement. Hvert inkrement kan bestå av tre iterasjoner:

1. I iterasjon 1 implementerer man de viktigste av de funksjonelle kravene på inkrementet. Dette kan sees på som en evaluering av designet på inkrementet.
2. I iterasjon 2 implementerer man den funksjonaliteten som gjenstår for inkrementet, og forbedrer de funksjonelle svakhetene man fant gjennom iterasjon 1.
3. I iterasjon 3 stabiliserer man inkrementet, slik at det kan kjøres en (del-)release av produktet for kunden.

Basert på evalueringer fra kunde/bruker, forbedrer man så inkrementene og legger til ny, ønsket funksjonalitet om man ser dette som hensiktsmessig. Det neste inkrementet som leveres kan dermed tilby forbedringer av det forrige inkrementet, pluss ny funksjonalitet. Systemet er komplett når all hensiktsmessig planlagt og foreslått funksjonalitet er bygget inn i produktet.

## 2.5 Avsluttende bemerkninger

I dette kapitlet er prosedyrer for endringshåndtering og de vanligste prosessmodellene blitt beskrevet, samt at evolusjonær, inkrementell utvikling er blitt eksemplifisert gjennom ulike instanser. De ulike prosessmodellene tar hensyn til sluttbrukerne og endringsforespørsler på ulike måter. Der fossefallsmodellen kun innlemmer brukerne i starten og slutten av utviklingen, prøver de inkrementelle og evolusjonære modellene å ta hensyn til kommunikasjon med sluttbrukere gjennomgående under utviklingen. Dette medfører at endringsforespørsler må håndteres under hele utviklingen. Evolusjonær, inkrementell utvikling håndterer endringer på en måte som gjør at godkjente endringsforespørsler implementeres i nye inkremitter av systemet, som igjen leveres til brukerne for en evaluering. På denne måten utvikles produktet mot et ferdig system.

Et viktig mål for evolusjonær og inkrementell utvikling er å identifisere kundens *virkelige og egentlige* behov ettersom systemet utvikles. Det bør her likevel understrekes at akkurat *hvordan* utviklingen burde gjennomføres for at man best kan oppnå de fordelene man håper på er et åpent forskningsspørsmål (Arisholm 2001). Det er plausibelt å anta at de til nå utviklede prosessmodellene trenger en betydelig andel av tilpasning basert på forskjellige karakteristika ved det spesielle utviklingsprosjektet man skal gi seg i kast med.





# Kapittel 3

## Metode

Dette kapitlet vil plassere denne hovedfagsoppgaven metodisk. Først i kapitlet er en beskrivelse av hvordan resultatene i denne oppgaven ble samlet inn, før innsamlingsmetodene beskrevet gis en teoretisk kontekst fra eksisterende litteratur.

### 3.1 Innhenting av materiale

Delkapittel 3.1 vil ta for seg hvordan resultatene for denne oppgaven ble samlet inn.

#### 3.1.1 Litteratursøk

Litteratursøket innen det overordnede temaet evolusjonær utvikling foregikk systematisk ved hjelp av tidsskrifter, bibliotekressursen BibSys, bibliografier og Internett. Resultatet av litteratursøket er presentert som bakgrunnsinformasjon i kapittel 2, samt som komparativ litteratur i kapittel fire og fem. Det bør merkes at to av artiklene referert til er whitepapers mens to andre artikler ennå ikke er publisert. Én referanse er også en avisartikkel. Noen få referanser henviser også til nettsteder. Informasjonen i de nevnte kildene er likevel ansett som pålitelig. Etter at litteratursøket var over, startet innhenting av empiri fra tre webutviklingsprosjekter. Kapittel fire i denne oppgaven fokuserer på denne erfaringsinnhenting. Her ble det foretatt intervjuer med 15 personer, bearbeiding av data, analyse og komparative litteraturstudier.

#### 3.1.2 Prosjektutvelgelse

Utviklingsprosjektene ble utvalgt på basis av relativt nylige prosjekter som Genera hadde vært involvert i, enten som hovedleverandør eller som underleverandør for en hovedleverandør. Siden erfaringsinnhenting er basert på enkeltindividens rekonstruksjon av hendelser, ble det antatt at det

Prosjektnavn	TelMont	BibTid	UniBase
Prosjektleder	Ja	Ja	Ja
Antall utviklere	2	2	1
Kunde	Ja	Ja	Nei
Antall brukere	4	2	Ingen

Tabell 3.1: Oversikt over personene som ble intervjuet

sannsynligvis ville øke kvaliteten på dataene å velge relativt nyavsluttede prosjekter/leveranser. Videre var målet å finne fram til relativt forskjellige prosjekter for å forsøke å avbilde så mange karakteristika ved webutviklingsprosjekter som mulig av både negative og positive erfaringer. Det ene prosjektet, det såkalte TelMont-prosjektet, bidro vesentlig mer til prosessrettingslinjene i kapittel fire enn de to andre prosjektene.

### 3.1.3 Personutvelgelse

Intervjuer med flere forskjellige personer involvert i webutviklingsprosjektene ble utført for å dekke flest mulige roller i utviklingsprosjektene, se tabell 3.1. 11 av disse intervjuene foregikk muntlig ved personlig oppmøte, mens fire av dem foregikk per e-post. Respondentene bestod av

- utviklere og prosjektledelse fra utviklerorganisasjonen,
- prosjektledelse hos kunde, og
- sluttbrukere.

### 3.1.4 Intervjuer

Foran hvert muntlig intervju ble flere spesifikke spørsmål formulert. Intervju-spørsmål kan være enten åpne eller lukkede (Faulkner 1998). I åpne spørsmål står respondenten fritt til å svare slik han måtte ønske, mens man i lukkede spørsmål begrenser respondenten til å svare fra en liste med ferdigformulerte svar. Ønsket var at respondentene skulle svare så fritt som mulig. Derfor ble åpne spørsmål valgt som spørsmålsform. Spørsmålene ble kombinert og prioritert slik at de til sammen dannet en intervjuguide. En intervjuguide består av en liste med relativt åpne spørsmål (Seaman 1999). Under intervjuene ble ikke intervjuguiden fulgt slavisk. Dersom enkelte temaer viste seg som spesielt interessante, ble mer detaljerte spørsmål stilt rundt dette. Hensikten med en intervjuguide å ha en huskeliste, slik at man får med de viktigste temaene som skal dekkes (Repstad 1998). Den samme intervjuguiden ble brukt mot alle respondentene, slik at de samme teamene ble berørt og alle fikk komme med sine synspunkter rundt det aktuelle emnet. Som et utgangspunkt

for datainnsamlingen fungerte intervjuguidene veldig bra. Andre, mer åpne spørsmål ble også stilt. Dette ble gjort for å fange opp uforutsett informasjon.

Blant brukerne som deltok i utviklingen av TelMont, ble det utført en e-postundersøkelse da disse var spredt geografisk utover hele landet. Et dokument med 13 ulike spørsmål ble sendt til brukerne på e-post. Det var god respons på undersøkelsen, fire av seks brukere tok seg tid til å svare. Svarene fra de fire brukere ble besvart separat – uavhengige av hverandre. Der data fra forskjellige brukere samsvarer med hverandre, er det dermed ikke fordi brukerne har påvirket hverandres svar.

De muntlige intervjuene ble spilt inn på kassettbånd for å unngå tap av informasjon. Det anbefales å bruke lydbåndopptaker ved forberedte, kvalitative intervjuer (Repstad 1998). En ulempe med kassettbåndopptak er at noen respondenter kan bli svært hemmet av båndopptakeren. Ingen av respondentene for denne oppgaven nektet at båndopptakeren ble brukt, men enkelte personer syntes tydelig det var mer ukomfortabelt enn andre. For det største og mest komplekse prosjektet, TelMont-prosjektet, ble alle spørsmål og svar fra opptakskassetten transkribert ordrett. Selv om dette var en detaljert og svært tidkrevende prosess som produserte omlag 100 sider tekstmateriale, var denne metoden viktig for å få med nøyaktigheten og rikdommen i intervjuene. Denne metoden var også viktig for å få med uforutsett informasjon – informasjon som var sammenflettet inn i flere ulike svar på spesifikke og åpne spørsmål. På TelMont-prosjektet ble det transkriberte materialet analysert i parallell av Erik Arisholm og forfatteren av denne oppgaven. Til slutt ble disse to separate analysene flettet sammen til en enhetlig analyserapport. Erfaringene fra denne analyseprosessen viste klart fordelene ved å utføre analysene parallelt. Dataene ble på denne måten analyserte ved hjelp av forskjellige perspektiver, noe som resulterte i ulike temaer og kategoriinndelinger. Hvis bare en person hadde analysert dataene, ville viktig informasjon gått tapt. Siden TelMont-prosjektet var det rikeste prosjektet for erfaringsinnhenting, ble det ansett som riktig å gjøre dette på den måten på dette prosjektet, men det var ikke anledning til å gjøre like omfattende analyse for det to andre prosjektene. Dette syntes det heller ikke være behov for, da disse var mindre detaljrike. For BibTid- og UniBase-prosjektene ble intervjuene fremdeles tatt opp på bånd, men ingen ordrett transkribering eller parallell analyse ble utført på samme omfattende måte. Derimot ble en intervjuanalyse laget for hvert intervju med hver respondent. Alle analyserapportene ble sendt til intervjuobjektene for at de skulle gjennomføre en kvalitetssikring og korrigerings av misforstått informasjon. Erfaringene som ligger til grunn for denne oppgaven er basert på de reviderte analyserapportene.

### 3.1.5 Data-analyse

De tre studerte prosjektene ble under analysen holdt separate, og det ble foretatt en *cross-case* analyse (Seaman 1999). En *cross-case* analyse innebærer å sammenligne de tre casene for å se etter likheter og ulikheter.

Under analysen fra erfaringsinnhentingene kom det særlig fram at endringshåndtering hadde vært et komplisert felt i prosjektene, hovedsakelig på grunn av prosjektenes evolusjonære karakter. Det virket derfor interessant å dykke nærmere ned i dette emnet, og en modell for endringshåndtering i evolusjonære prosjekter ble utarbeidet. Denne modellen ble først beskrevet ved hjelp av en prosessbeskrivelse, før den fikk sin konkrete manifestasjon i form av en verktøyprototype. Kapittel fem fokuserer på denne prosessbeskrivelsen, verktøyutviklingen og en verktøyevaluering. Metoder tatt i bruk i kapittel fem inkluderer læring av programmeringsteknikker, prototyping, åpen, passiv observasjon, bearbeiding av data, analyse og komparative studier.

### 3.1.6 Utvikling av prosessbeskrivelse og verktøyprototype

Siden verktøyprototypen skulle fungere over web, ble følgende teknologi brukt: UML-modellering, HTML-koding, CSS-stylesheets, scriptspråket Perl i kombinasjon med CGI-scripting og oppsett og bruk av en MySQL-database (Homer, Ullmann & Wright 1997, Schwartz & Christiansen 1997, Wall, Christiansen & Schwartz 1996, Castro 1998, Negrino & Smtih 1999, Axmark & Widenius 2000, Elmasri & Navathe 2000). Utviklingen av prosessbeskrivelsen og verktøyprototypen var i seg selv en evolusjonær prosess, der disse produktene ble utviklet på grunnlag av endringsønsker fra fem testbrukere — to prosjektledere i Genera, en ansatt hos bedriften SINTEF, en student ved Institutt for Informatikk og en prosjektleder hos bedriften Braathens. Endringsønskene kom fram gjennom åpne samtaler og observasjon av testbrukerne. I valget av testbrukere var det ønskelig å få representert ulike undergrupper av brukere. Det er viktig at testbrukerne er representative for flere typer brukere av en slik applikasjon.

Det ble først laget en enkel prosessbeskrivelse. Basert på denne enkle prosessbeskrivelsen startet implementeringen av skjermbilder i HTML. Da de første skjermbildene var oppe og kjørte som en del-leveranse, viste det seg naturlig å endre den initielle prosessmodellen. Den ble da videreutviklet og utbygd til å inkludere endringsforslag fra testbruker 1, 3 og utvikleren. Så ble det igjen gjort noe implementasjon basert på prosessbeskrivelsen. Dette medførte at prosessbeskrivelsen på nytt ble endret etter at man evaluerte neste del-leveranse. Etter de to del-leveransene, og tredje gjennomgang av prosessbeskrivelsen, lå prosessbeskrivelsen fast slik den er i sin nåværende form. Testbruker 1 trakk seg da noe ut av utviklingen, og en ny testbruker kom inn i utviklingen — testbruker 2. Den nye testbrukeren deltok aktivt til prototypen var ferdig implementert, sammen med testbruker 3 som hadde

deltatt også i utviklingen av prosessbeskrivelsen. De to testbrukerne evaluerte derimot prototypen uavhengig av hverandre, og hadde ikke noe innbyrdes kontakt.

Det komparative studiet på slutten av kapittel fem ble utført ved å laste ned demonstrasjonsversjoner av ulike verktøy for endringshåndtering fra Internett.

## 3.2 Metodevalg

Delkapittel 3.2 vil beskrive det teoretiske fundamentet bak forskningsmetodikken som er benyttet.

### 3.2.1 Metodeorientering

Resultatene presentert i denne oppgaven er bygd på en empiriske metoder. Järvinen skiller mellom to ulike tilnærminger til empiriske metoder: *teoritestende studier* og *teoriskapende studier* (Järvinen 1999). Teoristestende studier prøver å finne ut av om virkeligheten svarer til resultatene i en bestemt teori, mens teoriskapende studier ønsker å lage en teori, en modell eller et rammeverk som beskriver deler av virkeligheten. Denne oppgaven er både teoriskapende og teoristestende, men hovedvekt legges på den teoriskapende delen. Oppgaven ønsker å kartlegge en del av virkeligheten og bruke dette som en grobunn for praktiske retningslinjer som innspill til Genova web-prosess. Samtidig tester den også eksisterende litteratur opp mot resultatene som studiet ga.

### 3.2.2 Komparative litteraturstudier

En essensiell ingrediens i vitenskap er å relatere egne resultater med eksisterende litteratur (Järvinen 1999). Grunnen til at det er viktig å finne sammenlignbar litteratur som står i kontrast til egne resultater, er at det i motsatt fall er lett å tenke at forskeren har oversett denne litteraturen (Järvinen 1999). I så fall kan man få mindre tillit til resultatene som er funnet. Litteraturen i denne oppgaven er både i samsvar med og i opposisjon til resultatene.

### 3.2.3 Metoder for datainnsamling

Det finnes to overordnede metoder for datainnsamling: *kvantitative* og *kvalitative* metoder (Repstad 1998). Kvantitative data beskriver situasjonen som studeres ved hjelp av tallfestede variabler (Cornford & Smithson 1996). Kvantitative data gir dermed et lite antall opplysninger om mange undersøkelse-enheter (Dalland 1997). Et eksempel på kvantitativ metode er innsamling

av data ved hjelp av spørreskjemaer med predefinerte svaralternativer. Kvalitative data beskriver kompleksiteten i menneskelig oppførsel, og tar i større grad sikte på å fange opp mening og opplevelse som ikke lar seg tallfeste eller måle (Dalland 1997). I kvalitative studier er *opptelling* underordnet, i motsetning til i kvantitative studier. Kvalitative data representeres gjerne via bilder og ord, ikke numre alene (Seaman 1999). Kvalitative tilnæringsmåter beskriver nyansert “det som finnes”, og er mindre opptatt av hvor ofte det finnes. Fordelen med den kvalitative metode er at metoden tvinger forskerne til å dykke ned i kompleksiteten til et problem, slik at resultatene blir rikere og mer informative (Seaman 1999). Kvalitative data gir også mer forklarende informasjon enn kvantitative data. Denne oppgaven er hovedsakelig kvalitativ, og inneholder lite kvantitativt materiale.

### Case studier

En vanlig måte å samle inn kvalitativt materiale på, er gjennom *case studier*. Braa og Vidgen skiller mellom to ulike former for case studier — *hard case studier* og *soft case studier* (Braa & Vidgen 1999). Et hard case studium har sitt grunnlag i en positivistisk forskningstradisjon. I den positivistiske forskningstradisjonen er forskeren en observatør som har som miljø et laboratorium (Braa & Vidgen 1999). Problemstillingen starter med et “hvordan” eller “hvorfor”, og virkeligheten beskrives gjennom tallfestede, analyserte data, noe som gir hard case studiene et kvantitativt preg (Arisholm 2001). Som kontrast finner man en interpretativ forskningstradisjon, og her har soft case studiet sine røtter (Braa & Vidgen 1999). Interpretative forskere har som utgangspunkt at tilgang til virkeligheten skjer gjennom sosiale konstruksjoner som språk, bevissthet og delt mening (Walsham 1993, Myers 1997). I interpretative studier ønsker man å tolke og forstå situasjoner der mennesker er aktører, og dataene som beskrives er ofte kvalitative (Braa & Vidgen 1999). I en interpretativ forskningstradisjon mener man at naturvitenskapens metoder er utilstrekkelige når det gjelder studier der mennesker er involvert, mest fordi ulike mennesker vil tolke en situasjon på ulike måter (Braa & Vidgen 1999). Det er dermed vanskelig å generalisere og lage generelle lover. Denne oppgaven er interpretativ og et typisk soft case studium der intervjuresultater tolkes og forsøkes forklart. Et alternativ kunne ha vært å utføre et *eksperiment*. Et eksperiment frigjør et fenomen fra dets kontekst fordi konteksten er “kontrollert” av et laboratoriemiljø, slik at oppmerksomheten kan vies noen få variable (Yin 1995). Et eksperiment ble ikke utført i denne oppgaven fordi problemstillingen ikke var kjent på forhånd, men ble funnet eksplorativt gjennom innsamling av materiale. Tidsrammene rundt oppgaven ble dermed begrensende.

Generelt er et case studium en empirisk undersøkelse som utforsker et fenomen innenfor en virkelig kontekst, særlig når grensene mellom fenomenet og konteksten ikke er selvinnlysende (Yin 1995). Et case studium gir en inngåen-

de forklaring der man søker å få en helhetsforståelse av situasjonen (Cornford & Smithson 1996). Et case studium er særlig hensiktsmessig dersom man ønsker å beskrive også kontekstuelle faktorer som kan være relevante for resultatene (Yin 1995). Et case studium tillater at virkeligheten kartlegges i detalj, og at mange ulike variabler kan analyseres (Arisholm 2001). I et case studium er det ikke mulig å replikere dataene via et formelt eksperiment (Kitchenham, Pickard & Pflieger 1995). Resultatene i denne oppgaven er basert på tre case studier der det er sannsynlig at konteksten har innvirkning på resultatene. Konteksten er derfor beskrevet for å øke forståelsen av studiene. Muligheten til å fordype seg gjør at et case studium kan være en god måte å etablere nye teorier, eller teoriskapende forskning. Derimot kan man ikke ut i fra et case studium generalisere fordi et slikt studium er så kontekststøttet. Ofte er det derfor interessant å sammenligne med andre tilsvarende caser (Kitchenham et al. 1995, Repstad 1998). En slik sammenligning er gjort i denne oppgaven ved hjelp av å sammenligne de tre casene med hverandre, og så å sammenligne dem med eksisterende litteratur.

## **Intervjuer**

En vanlig måte å samle kvalitative data fra case studier på er gjennom intervjuer (Seaman 1999). Seaman skiller mellom to typer intervjuer: *strukturerte* og *ustrukturerte* intervjuer. I et strukturert intervju er spørsmålene formulert på forhånd og respondenten besvarer disse. I et ustrukturert intervju er hensikten at både spørsmål og svar skal komme fra respondenten (Seaman 1999). I et ustrukturert intervju vet ikke intervjueren på forhånd hva slags informasjon som vil bli gitt. Hensikten er å skaffe så mye informasjon som mulig om et bredt definert emne. En vanlig form for å skaffe informasjon i kvalitative studier er å blande disse to intervjuformene – såkalt semistrukturert intervjuteknikk (Seaman 1999). Semistrukturerte intervjuer inneholder en blanding av åpne og spesifikke spørsmål, og tillater at man lager nye spørsmål på sparket dersom uforutsett informasjon skulle dukke opp. Det var en slik framgangsmåte som ble brukt under de muntlige intervjuene som ble gjort i denne oppgaven. Under e-postintervjuene ble det derimot brukt en strukturert intervjuform, der ti predefinerte spørsmål ble sendt til de ulike respondentene.

## **Tekstanalyse**

Analyse av kvalitative data er en tidkrevende og omstendelig prosess (Halvorsen 1993), fordi kvalitative data består av en stor mengde ustrukturert informasjon, i dette tilfellet omlag hundre maskinskrevne sider transkribert materiale, samt 12 ulike intervjuanalyser. Analysen for kvalitative data er heller ikke så formalisert som analyse av kvantitative data, og skjer mer intuitivt med utgangspunkt i de foreliggende tekster. Kvalitativ analyse

er også mer personlig siden det ofte er den som har samlet inn dataene som analyserer dem (Repstad 1998).

Halvorsen nevner to former for tekstanalyse — helhetsanalyse og delanalyse. Delanalyse forutsetter at det tekstlige materialet kan deles opp i ulike kategorier. For å få fram meningsinnholdet i tekstmaterialet bør det i rapporten siteres fra feltobservasjonene (Halvorsen 1993). Problemet med en slik analyseform er at leseren ikke vet hvor representative enkelte sitater er. En helhetsanalyse konsentrerer seg i stedet om et allment inntrykk forskeren har fått ved gjennomlesning av tekstmaterialet. Deretter prøver forskeren å velge ut situasjoner som illustrerer helhetsinntrykket. Ved denne framgangsmåten risikerer man å gå glipp av en variasjon og kompleksitet. Analysen av de tre studerte prosjektene er en kombinasjon av helhetsanalyse og delanalyse. Temaer er kategorisert i en delanalyse med dertil hørende sitater, før det gies en mer overordnet helhetsforståelse for inntrykket fra felldataene og sammenligningen av de ulike prosjektene.

### 3.2.4 Prototyping

For å utvikle verktøystøtten for denne oppgaven, ble det evolusjonært lagd en prototype. Prototyping er detaljert beskrevet i kapittel 2. Verktøyprototypen utviklet i denne oppgaven er et eksempel på en *eksperimenterende prototype* (Floyd 1984). Prototypen er foreslått som en løsning på et identifisert problem — problemet med ustrukturert endringshåndtering i evolusjonære utviklingsprosjekter. Metoden som ble brukt var i utgangspunktet full funksjonell simulering, men etter eksperimenterende bruk med prototypen er det oppdaget ny funksjonalitet som burde inkluderes i en endelig versjon av prototypen. I prototypen er det ikke vektlagt noen ikke-funksjonelle krav, og kildekoden som er implementert er ikke egnet for bruk i et kommende verktøy for endringshåndtering.

### 3.2.5 Observasjon

Observasjon er å studere en respondent i dens eget miljø, og på denne måten samle inn data systematisk (Seaman 1999). Ved skjult observasjon er den eller de som observeres som oftest uvitende om at de blir observert — man skaffer seg normalt en generell tillatelse til å observere fra for eksempel de observertes ledelse, men oppgir ikke noe tidspunkt for observasjonen eller hvem man ønsker å observere. Ved åpen observasjon foreligger en tillatelse fra de spesifikke personene som skal observeres før en går i gang med observasjonen, og man avtaler et observasjonstidspunkt (Halvorsen 1993, Dalland 1997). Man forteller likevel ikke de som observeres i detalj om hvilke problemstillinger en er i ferd med å belyse (Repstad 1998).

Under utviklingen av verktøyprototypen, ble åpen observasjon praktisert. Observasjonen kan være *deltakende* eller *passiv* (Dalland 1997, Repstad 1998,



Halvorsen 1993). Ved deltakende observasjon går man selv inn i det feltet som skal studeres og praktiserer en oppgave. En passiv observatør deltar ikke i oppgaver, men prøver å virke minst mulig inn på det felt eller de oppgaver som gjøres. Observasjonen gjort for denne oppgaven bærer preg av å være passiv observasjon. Observatøren deltok ikke i oppgaver, men forholdt seg passiv da testbrukerne testet ut verktøyprototypen. For å lette på stemningen, ble det gjennomgående småpratet om ting som ikke vedrørte verktøyet. Dersom testbrukerne lurte på noe, ble enkelte ting forklart underveis. Som Repstad påpeker, kan det være et problem hvis den som observerer sitter helt taus. Personene som observeres kan da føle seg truet fordi den som observerer ikke gir noe av seg selv. En total passivitet kan skape usikkerhet og kanskje aggresjon hos aktørene (Repstad 1998). Det er viktig at den som blir observert ikke tenker uavbrutt over at den blir observert (Seaman 1999). Et problem ved å fortelle for mye til aktørene som Repstad nevner, er at de observerte lett kan ønske å “snakke forskeren etter munnen”. Under småpraten ble det derfor ikke holdt foredrag om verktøyprototypen og om hensikten med den, men samtalen ble dreid mer i retning av trivialiteter — i hensikt av å løse opp stemningen og få de observerte til å snakke mer.

Under observasjonene ble det ikke lagd noen skriftlige notater for å registrere feltdataene. Derimot ble det laget et referat rett etter møtene av hvordan observasjonen hadde vært. Grunnen til at notatene ikke ble gjort under observasjonen, var en antakelse om at flittig notering ville være forstyrrende for testbrukerne og binde dem unødvendig. Det kan være metodisk fornuftig å ikke notere mens man observerer, fordi notering fra forskerens side kan gjøre at de som observeres lar det gå sport i å teste ut hva som vekker forskerens interesse (Repstad 1998). Feltnotatene som ble lagd etter observasjonene ble så brukt som grunnlag for videre iterasjoner på verktøyet.

En fare ved å bruke observasjon som en evalueringsmetodikk, er at det antas at mennesker som observeres vil oppføre seg annerledes når de vet at de blir observert (Dalland 1997, Repstad 1998, Halvorsen 1993). Erfaring tilsier likevel at man etter kun kort tid lar seg påvirke av at man blir observert. Siden framgangsmåten ved observasjon også sjelden er standardisert, er også reliabiliteten ved slik observasjon lav (Halvorsen 1993). Det er dessuten vanskelig å si hvor representativt utvalget av testbrukerne er.



## Kapittel 4

# Erfaringer fra webutviklingsprosjektene

Dette kapitlet beskriver prosessretningslinjer rundt endringshåndtering basert på en erfaringsinnsamling fra tre webutviklingsprosjekter Genera har vært involvert i. Anbefalingene er ment for å gi innspill til Genova webprosess, og ønsker å besvare problemstilling 1:

***Problemstilling 1:** Hvilke generelle erfaringer fra evolusjonær webutvikling kan være nyttige for andre evolusjonære webutviklingsprosjekter, og hvordan kan retningslinjene formaliseres i Genova webprosess?*

Først i kapitlet er en detaljert beskrivelse av utviklingsprosjektene som er studert. Deretter følger prosessretningslinjene som ble funnet ut i fra analysene som ble gjort. Prosjektet TelMont bidro vesentlig mer til erfaringsinnhenting enn de andre prosjektene fordi dette var det største prosjektet. TelMont er derfor beskrevet mer detaljert enn BibTid og UniBase.

### Prosjektoversikt

En oversikt over prosjektene er gitt i tabell 4.1 på neste side. De tre prosjektene er relativt ulike i bruksområde og utviklingsmetodikk. Felles for de tre prosjektene er at de ble utviklet av en leverandør for en ekstern kunde og at produktene som ble utviklet alle er webapplikasjoner som prosessmodellkjøres på et intranett.

Prosjektene er beskrevet i henhold til følgende mal:

- Prosjektorganisering
- Prosjektaktiviteter og milepæler
- Estimerer
- Prosessmodell

Prosjektnavn	TelMont	BibTid	UniBase
Beskrivelse	Webbasert støttesystem utviklet for et norsk telekommunikasjonsfirma. Utfører komprimering og optimalisering for ISDN-kort.	Tidsskriftsformidling for biblioteker. Tilbyr søk etter tidsskrifter, samt opprettelse, oppsigelse og vedlikehold av abonnementer.	Genererer økonomiske og administrative rapporter fra ulike databaser tilhørende universiteter og høyskoler i Sverige.
Antall utviklere	10-15, pluss noen eksterne konsulenter	3	4
Kontraktstype	Ekstern leverandør, timebasert	Ekstern leverandør, fastpris	Ekstern leverandør, fastpris
Varighet	9 måneder	6 måneder	5 måneder

Tabell 4.1: Oversikt over de tre webutviklingsprosjektene

- Kravspesifikasjon
- Prototyping
- Endringshåndtering
- Brukermedvirkning

## 4.1 TelMont

TelMont er et webbasert støttesystem som skal brukes av et norsk telekommunikasjonsfirma. Brukerne av systemet er profesjonelle ingeniører. Systemet komprimerer plass i telefonsentraler for at man senere skal kunne montere nye ISDN-kort som tar mindre plass. Mye vekt er blitt lagt på forenkling av arbeidsprosessen for dette formålet, det vil si på teknisk gode løsninger og tids- og plassbesparing. Seks brukere har vært aktivt deltakende i dette. En oversikt over hovedtrekkene av erfaringene fra TelMont er vist i tabell 4.2 på neste side.

### 4.1.1 Prosjektorganisering

Det var et eksternt firma som utviklet TelMont-produktet for kundeorganisasjonen, og konsulenter fra Genera ble leid inn i dette arbeidet. Prosjektet besto av omtrent 10-15 personer, og i tillegg konsulentene fra Genera. Spesialister i kundeorganisasjonen som kjente til arbeidsflyten rundt

Aktivitet	Hovedpunkter	Beskrevet under
<b>Estimering</b>	Estimatene ble overskredet med 100%. Estimatoverskridelsene tilskrives komplisert forretningslogikk, ukjent teknologi og særlig manglende kontroll over endringshåndteringen.	4.1.3 på side 43
<b>Prosessmodell</b>	I startfasen en evolusjonær, intern modell kalt "løsningsleveransesmodellen". Underveis i prosjektet forandret leverandør prosessmodell til fossefallsmodellen.	4.1.2 på side 43 og 4.1.4 på side 45
<b>Kravspesifikasjon</b>	En rekke endringer på kravspesifikasjonen. Skyldtes interne stridigheter i startfasen, sen brukerinvolvering og en rekke endringsønsker. Kundeorganisasjonen ønsket at produktet skulle være mest mulig fleksibelt med muligheter for senere utvidelser. Brukerne syntes den initiale kravspesifikasjonen var for kompleks og at den tok høyde for mye unødvendig funksjonalitet de ikke var interesserte i. Resultatet ble en betydelig mengde endringsønsker fra brukerne, omgjøringsarbeid og forenkling av kravspesifikasjonen.	4.1.5 på side 45
<b>Prototyping</b>	Prototypen ble tillagt atskillig mer funksjonalitet enn hva som opprinnelig var tanken. Dette gjorde det ble nærmest umulig å forsvare utskifting av teknologi.	4.1.6 på side 46
<b>Endringshåndtering</b>	Det ble produsert et stort antall endringsønsker som ikke alltid ble håndtert gjennom formelle kanaler. Prosjektleder hos både kunde og leverandør følte de mistet kontroll over utviklingen, særlig fordi kravspesifikasjonen ikke ble oppdatert i takt med forandringene som endringsønskene medførte.	4.1.7 på side 46
<b>Brukermedvirkning</b>	Alle parter var svært fornøyde med brukermedvirkningen. Brukerne mener også at endringsforslagene deres gjorde produktet mer tilpass et deres behov. Brukerne beskriver det som vanskelig å se for seg funksjonalitet, men synes prototyper var en svært god hjelp.	4.1.8 på side 47

Tabell 4.2: Oversikt over erfaringer fra prosjektet TelMont

ISDN-komprimering og optimering bidro til analyse og design. Brukerne på TelMont var de kommende brukerne av applikasjonen, valgt ut fordi de både hadde ulike arbeidsfunksjoner i organisasjonen og fordi de befant seg geografisk spredt rundt i landet. På TelMont var ikke brukerne og kunden sammenfallende. Prosjektet ble ikke lagt ut på anbud, og faktureringen var timebasert (det vil si ikke fastpris). Vedlikehold av systemet skulle utføres av utviklerorganisasjonen.

#### **4.1.2 Prosjektaktiviteter og milepæler**

##### **Mai 1999 - August 1999: Forprosjekt.**

Forprosjektet startet opp i regi av kundeorganisasjonen i mai 1999. Formålet med forprosjektet var å vurdere behovet for en slik automatisert løsning. Forprosjektet involverte en person fra utviklerorganisasjonen og 3-4 personer fra kunde. Leveransen fra forprosjektet var en analyserapport.

##### **August 1999 - Oktober 1999: Utarbeidelse/Retningsfase.**

På sensommeren startet arbeidet med å lage kravspesifikasjon og delvis en analyse. Leverandør besluttet å bruke en evolusjonær prosessmodell på prosjektet, en intern modell kalt "løsningsleveransemodellen" som organisasjonen hadde erfaring med fra tidligere prosjekter. Løsningsleveransemodellen innebar inndeling i tidsbokser på omlag 6 måneder, hvor hver boks ble avsluttet med en leveranse (reell eller verifikasjonsleveranse). I denne fasen ble de overordnede prosessene som TelMont-systemet skulle støtte diskutert (komprimering, kabelomlegging og speiling). Arbeidet resulterte i en kravspesifikasjon og en analysemodell som bestod av et flytskjema og noen Use-Cases samt en enkel prototype av skjermbilder.

##### **Oktober 1999 - Mars 2000: Iterativ utarbeidelse og konstruksjon (Analyse/Prototyping/teknologieuvaluering.)**

I denne fasen av prosjektet begynte utviklerorganisasjonen med en omfattende evolusjonær prototyping. Prototypingen hadde to formål:

- å evaluere hvorvidt den valgte teknologien var gjennomførbar, og
- å detaljere kravspesifikasjonen gjennom å skaffe et detaljert brukergrensesnitt over den viktigste funksjonaliteten.

Allerede i september 1999 begynte utviklere fra utviklerorganisasjonen å lage en enkel prototype i HTML for å beskrive skjermbilder og arbeidsprosesser. Samtidig startet et større prototypingprosjekt som hadde til hensikt å teste ut teknologien som var valgt. Den skulle dessuten detaljere noen av skjermbildene og det grafiske brukergrensesnittet som kunden hadde spesifisert. I desember startet designfasen, og produktet ble modellert i UML. Modellen ble lagt åpen på en lokal server slik at kunden, og senere brukerne, hadde tilgang til den hele tiden under utviklingsprosessen. Første prototype,

som fungerte som en arkitektur-release, var ferdig i mars 2000. Før den var ferdigstilt, ble det arrangert en workshop for å evaluere skjermbildene og arbeidsflyten.

Allerede i oktober/november begynte utviklerorganisasjonen å etterlyse brukerne. De ville ha reelle brukere, men disse ble ikke skaffet før i slutten av januar/begynnelsen av februar. Problemet var at brukerne var viktige ingeniørressurser i kundeorganisasjonen, og det ble mye diskusjon mellom forskjellige grupper innad i kundeorganisasjonen om hvorvidt og når disse ressursene skulle delta i TelMont-prosjektet.

#### **Mars 2000: Endret Prosess.**

Prosjektet var nå forsinket, og for å bli ferdige mente prosjektleder hos utviklerorganisasjonen at det var nødvendig å gjøre en omlegging fra den iterative prosessen (løsningsleveransemodellen) til en fossefallsprosess. Han krevde nå formell akseptering av detaljspesifikasjoner og av kode på metode-nivå. Denne radikale endringen av prosessmodell ble sett på som en måte å få økonomi- og tidsplaner under kontroll. Utviklerorganisasjonen mente at de ikke lenger kunne forsvare å bruke løsningsleveransemodellen på grunn av store forsinkelser og mange klager fra kunden. Klagene fra kunden skyldtes at prototypingen hadde tatt mye mer tid enn forventet, og den iterative prosessen førte til at det kom endringer hele tiden. Kunden opplevde nå stor frustrasjon fordi de følte de mistet kontroll på grunn av at kravspesifikasjonen ble endret kontinuerlig, uten at den formelle dokumentasjonen ble endret i samsvar med de nye endringene. De følte seg også forbigått i beslutningsprosessen rundt hvorvidt endringsønskene skulle implementeres, og dette endte i store diskusjoner. Det bør understrekes at utviklerne hadde sterk tiltro til at endringsønskene fra brukerne var ytterst nødvendige for å få til et realistisk og nyttig produkt. Kommunikasjonen rundt endringshåndteringen sviktet derimot grunnleggende mellom kunde og utviklerorganisasjon.

#### **Mai 2000: Endelige detaljspesifikasjoner godkjent.**

#### **Juni 2000: Første leveranse.**

Første leveranse var i juni, og bestod av et fullt operativt system. Prosjektet hadde da tatt tre måneder lengre enn estimert. Prosjektet hadde dessuten brukt om lag 8600 timer framfor de estimerte 4300 timene. Brukerne var *svært* fornøyde med det ferdige produktet.

### **4.1.3 Estimering**

For estimering av tidsforbruk brukte prosjektet funksjonspoengestimering. Erfaringstall fra andre webprosjekter hos leverandør ble brukt som grunnlag for estimeringen. Overskridelser i timeforbruk i forhold til estimatet for utviklingsprosjektet var omtrent 100%. Disse overskridelsene kan skyldes flere faktorer. Det viste seg at erfaringstallene ikke var så relevante som antatt.

Blant annet var det mye mer forretningslogikk i TelMont enn i de systemene som erfaringstallene var basert på, som var mer rene presentasjonssystemer samt litt logikk. Dette kan forklare en del av kostnadsoverskridelsene i forhold til estimatene for prosjektet. I TelMont brukte man mye mer tid enn antatt på forretningslogikk. I tillegg var mye av forretningslogikken implementert i ny teknologi, og det var i disse delene av koden overskridelsene var absolutt størst. Integrasjonen mellom teknologi fra ulike produsenter fungerte heller ikke bra. Basert på intervjuene med utviklerne er det liten tvil om at problemer med teknologien og mangel på kompetanse på teknologien ved oppstart medførte lavere produktivitet enn man kanskje ellers ville forventet.

Prosjektlederen hos kunde var ikke fornøyd med håndteringen av estimatene hos leverandør. Først når leverandør så at det var betydelige tidsoverskridelser, ble det utarbeidet et nytt estimat. Prosjektleder hos kunde mente dette estimatet burde kommet *før* man så at det var overskridelser. Han mener årsaken til estimatoverskridelsene var *“mye på grunn av teknologivalg, og endringer her og der som ble snikinnført.”* Dessuten mener han at leverandør ikke satte foten ned godt nok når man merket at det ble overskridelser.

*“Man gjorde ikke noe med prosjektet, man lot det skure og gå. Man fikk vite gjennom fakturaer som kom nesten en måned for seint, at her har vi hatt en overskridelse. Så blir det holdt et krisemøte hos leverandør, der de ringer oss og forteller akkurat det samme som vi fikk faktura på for noen dager siden. Ikke helt bra måte å følge opp på...”*

Prosjektlederen hos kunde tror ikke estimatet ble tatt seriøst nok av leverandøren fordi det var et timebasert prosjekt. Han mener også at de sammen burde definert målene og milepælene for prosjektet strammere og klarere. Han mener det var lite kommunikasjon mellom systemutviklere og kunde om når og hvilke milepæler som var passert.

Leverandør mente på sin side at en god del av overskridelsene skyldes at kunden stadig kom med endringer i krav. Kravene var vage, og det tok lang tid og mye ressurser før de reelle kravene til systemet ble identifisert. Dette skyldes blant annet sen tilgang på reelle brukere. Leverandør mener også overskridelsene skyldes at det ble også tatt i bruk mye ny teknologi på dette prosjektet. I den forbindelse fungerte prosjektet som kompetanseoppbygging for utviklerne.

I følge leverandør fikk iterativ utvikling og prototyping negative økonomiske konsekvenser. Leverandør tapte økonomisk på at alle endringene ikke ble godt nok dokumentert for å forsvare overskridelsene i tidsforbruket overfor kunde. Det bør i denne sammenheng understrekes at iterativ utvikling og evalueringen av prototypene bidro til at kunden fikk et *godt* produkt.



Det har vært lite feilretting, og endringene som kommer er lette å implementere. Brukerne er meget fornøyde. Kunden fikk også et *riktig* produkt. Samarbeidet med brukerne medførte at man innså at man ikke trengte å ta høyde for masse unødvendig “fleksibilitet” som ikke grunnet i brukernes reelle behov. Produktet ble antagelig lettere å bruke på grunn av den iterative utviklingsmodellen som tillot endringsforslag fra brukerne underveis. Derfor kan man argumentere for at ressursene som ble lagt ned på implementasjon og evaluering av prototyper ikke var så kostbare som regnearket tilsier.

#### 4.1.4 Prosessmodell

Som nevnt i 4.1.2 på side 43 benyttet prosjektet i startfasen den evolusjonære “løsningsleveransemodellen”, men underveis i prosjektet forandret leverandør prosessmodell til fossefallsmodellen på grunn av problemene relatert til prosjektstyring og kostnadsoppfølging.

#### 4.1.5 Kravspesifikasjon

Denne initielle kravspesifikasjonen ble utarbeidet av kunde i samarbeid med leverandør, og prosjektleder fra kunde var i denne perioden ansvarlig for kravspesifikasjonen. Å skrive den initielle kravspesifikasjonen var ifølge prosjektleder hos kunde “*som å slippe en tolvåring ut i det voksne livet.*” fordi prosjektmedlemmene ikke hadde oversikt over hvilke detaljer det var viktig å tenke igjennom ved oppstart av denne typen IT-prosjekter. Han beskriver denne prosessen som rotete og konfliktfylt, særlig fordi det var en del interne stridigheter i kundeorganisasjonen rundt konseptvurderingen og rundt hvordan det ferdige produktet burde bli. Han mener at dette medførte at mange av de endringene som senere kom opp var nødvendige.

Brukerne kom med i utviklingen et halvt år etter arbeidet med kravspesifikasjonen var påbegynt. Brukerne hadde en helt annen oppfatning enn kunderepresentantene av hvilke krav systemet burde ha. De syntes den initielle kravspesifikasjonen var for kompleks og at den tok høyde for mye unødvendig funksjonalitet de ikke var interesserte i. Resultatet ble at utviklere og brukere sammen gjorde en betydelig mengde omgjøringsarbeid og forenkling av kravspesifikasjonen. Både kunde og utviklere mener at brukerne bidro til å gjøre en rekke viktige nedskjæringer og endringer på den initielle kravspesifikasjonen i denne perioden.

Både kunde, leverandør og brukerne selv syntes i ettertid brukerne kom for seint inn i prosessen med å skrive kravspesifikasjon. Fra brukerne kom det fram at de kunne hatt større påvirkning på produktet dersom personer med rett fagkompetanse hadde vært involvert tidligere. De mener en tidligere involvering ville vært både tids- og kostnadsbesparende for kunden. Utviklerne mener den største ulempen ved sen brukermedvirkning var at brukerne

ikke hadde noe eierskap til kravspesifikasjonen fordi de kom inn så seint, og dermed ble fremmed overfor den. Dette gjorde at de i en startfase hadde problemer med å kritisere den. Utviklerne synes det tok for lang tid før brukerne torde å si ifra om krav som var feilspesifisert, unødvendige eller manglende. Før denne meningsytringen kom fra brukerne brukte utviklerne mye tid på å legge unødvendig kompleksitet inn i produktet. Brukerne er på sin side samstemte på at de i oppstarten synes det var vanskelig å delta. En av brukerne forteller at det var veldig mye papirer med diverse spesifikasjoner, samt at han til tider ikke følte seg datakyndig nok til å følge med i fagspråket. Da brukerne begynte arbeidet med kravspesifikasjonen var det også mange og lange diskusjoner der det for dem føltes som om de stod helt stille. En bruker forteller det slik:

*“Møtene i oppstarten var av en karakter jeg aldri tidligere hadde opplevd. Detaljene var mange og enkelte sitt markeringsbehov var stort. Jeg viser stor forståelse for at detaljer ved produktet er viktig, men synes at arbeidstiden til brukere er for verdifull til å sitte å krangle om detaljer.”*

#### **4.1.6 Prototyping**

Det tok atskillig lenger tid å utvikle prototypen enn først antatt, og den ble tillagt en del mer funksjonalitet enn hva som opprinnelig var tanken. Den delen av prototypen som skulle evaluere arkitekturløsningen ble svært bred, med mye funksjonalitet basert på en komplisert og omfattende kravspesifikasjon. Siden mye av prototypingen var basert på den opprinnelige kravspesifikasjonen — som ble skåret drastisk ned da brukerne kom med — måtte også mye av arbeidet med prototypen gjøres om.

Den lange tidsbruken førte til at prototypen senere ble utvidet til å bli et fullt operasjonelt system. Det var diskusjoner rundt om man skulle forkaste teknologien, men det ble nærmeste umulig å forsvare utskifting av teknologi på grunn av alt arbeidet som var lagt ned i prototypen. Man beholdt da den valgte teknologien selv om den hadde åpenbare mangler.

#### **4.1.7 Endringshåndtering**

Siden både kunde og brukere gjennomgående hadde tilgang til systemene, kunne de teste løsninger og foreslå endringer kontinuerlig. Arbeidsmøtene og den kontinuerlige evalueringen av webløsningen som brukerne gjorde, førte til at det ble produsert et stort antall endringsønsker som ikke alltid ble håndtert gjennom formelle kanaler. Det vil si at endringsønskene ble diskutert mellom brukere og utviklere, mens prosjektledere hos både kunde og leverandør ofte ble utelatt fra beslutninger rundt endringsønskene. Dette førte til at prosjektlederne følte de mistet kontroll over utviklingen, særlig

fordi kravspesifikasjonen ikke ble oppdatert i takt med forandringene i koden som endringsønskene medførte. Som prosjektleder var det dermed vanskelig å holde kontroll over utviklingen og pengeflyten. Prosjektlederen hos kunden var *svært* misfornøyd med endringshåndteringen hos leverandør. Han mislikte særlig at endringsønsker ikke ble sendt til ham for godkjenning før de ble lagt inn i produktet. Han hadde ønsket seg en mer kontrollert prosess for endringshåndtering. Kundens prosjektleder mener at en stor del av estimatoverskridelsene skyldes for dårlig fokus på endringsstyring hos leverandør underveis i prosjektet.

*“Endringer var ikke flagga på forhånd av mellomledelsen, og ingen hos oss var i stand til å fortelle hva det her kosta. Estimater var rett og slett ikke helt tatt rede på. Man tenkte ikke estimat når man tenkte endring.”*

Utviklerne ga klart uttrykk for at det å oppdatere kravspesifikasjonen i takt med alle endringene ville være en umulig oppgave. De prioriterte i stedet å kode siden prosjektet allerede var forsinket. Men utviklerne mente samtidig at en god dokumentasjon av endringene hadde spart dem mange diskusjoner med kunde. Manglende dokumentasjon av endringsønskene førte til at det hele endte opp i en pengediskusjon da tidsforbruket oversteg estimatet. Her bør det merkes at samtlige prosjektdeltakere mener at systemet ikke ville fungert tilfredsstillende for brukerne hvis man ikke hele tiden hadde hatt disse endringene og forbedringene i kravene.

#### **4.1.8 Brukermedvirkning**

Alle parter var svært fornøyde med at brukerne deltok i utviklingen. Både kunde og leverandør understreker at produktet ikke ville blitt på langt nær så bra og riktig dersom brukerne ikke hadde vært involvert i utviklingsprosessen. Utviklerne hadde et godt samarbeid med brukerne, og de hadde inntrykk av at brukerne var fornøyde. De beskriver brukerne som positive og interesserte i å bruke tid på prosjektet.

Å delta som bruker i TelMont krevde svært mange arbeidstimer. Fordi prosjektet hele tiden hadde lite personalressurser, mener én bruker at dette har gjort at trykket på den enkelte ble for stort. Flere av brukerne syntes også det var slitsomt at de ikke fikk noen form for avlastning i deres daglige jobb. Brukerne på TelMont har forskjellige erfaringer rundt hvorvidt de fikk aksept fra sin ledelse for å delta i utviklingen. En av brukerne sier at hans ledelse var positivt innstilt fordi de mente at muligheten til å påvirke og til å lære produktet ville være størst ved å delta. En annen bruker forteller derimot at hans utgangspunkt for å delta i prosjektet var at jobben hans måtte komme i første rekke. Beskjeden han fikk fra ledelsen var at: *“Bare jobben blir*

*gjort, må du gjerne delta i prosjektet*". En siste bruker forteller at han måtte overtale ledelsen til å få delta i prosjektet.

Flere av brukerne beskriver som vanskelig å tenke ut alle eventualiteter og alternativer som burde være med i produktet, og å forestille seg hvordan produktet ville se ut idet de foretok valg. Det var vanskelig å se for seg god funksjonalitet, men skjermbilder som ble presentert underveis var en god hjelp til å visualisere. Det beskrives også som krevende å huske alle viktige funksjoner systemet burde ha, og de synes det var vanskelig å være sikre overfor utviklerne at de ikke hadde glemt vesentlige elementer.

En bruker sammenligner prosessen med å bygge et hus, der man også spør seg hvordan tegningen av huset vil bli seende ut i det virkelige liv. Utviklingen ble – som med husbygging – lettere etter hvert som noe kom på plass og produktet ble mer synlig.

Alle brukerne som er intervjuet tror udiskutabelt at endringsforslagene brukerne kom med var nyttige for at produktet ble mer tilpasset brukerne. De anbefaler også andre brukere å delta i en utviklingsprosess. Fordeler ved brukermedvirkning som nevnes av brukerne er:

- Produktet ble enklere å bruke etter at utviklerne ble forklart arbeidsprosessene til en montør.
- Brukerkonsultasjonen bidro til at oppbygging av bilder, bildene sitt forhold til hverandre og logisk framdrift i programmet har blitt god. Brukerne har også fått positive tilbakemeldinger fra andre brukere på at TelMont er enkelt å lære og lett å bruke.
- Brukermedvirkningen var nødvendig for å oppfylle minstekrav til funksjonalitet.
- Brukerne av systemet har den riktigste forutsetningen til å fremme gode løsningsforslag til produktet. Tidligere har brukerorganisasjonen opplevd at utviklingen av et system feilet fullstendig fordi brukerne ikke ble spurt til råds før det var for sent. Dette hadde det vært svært enkelt for brukerne å si ifra om hadde de blitt konsultert under utviklingen.

Brukerne synes endringsforslagene deres ble godt mottatt. Endringsforslag måtte ha en god begrunnelse, men ble tatt på alvor. Endringer og forslag til tilpasninger ble møtt med åpenhet av leverandør, og diskutert nøye. Det var flere møter underveis der brukerne diskuterte utviklingen av produktet, og her synes brukerne at deres vurderinger om fagområdet ble tatt på alvor. Alle brukerne tror udiskutabelt at endringsforslagene brukerne kom med var nyttige for at produktet ble mer tilpasset dem som brukere.

Brukerne på TelMont er fulle av positive kommentarer til hvordan de synes det ferdige resultatet er blitt. Alle brukerne er svært fornøyde med produktet,

og forteller at flere andre brukere de har snakket med også synes produktet fungerer svært godt. En bruker sier om produktet at:

*“det er lett å lære, lett å bruke og det letter denne typen arbeidsoppgaver mye. Å kunne flytte abonnemeter uten å måtte tenke på kundeprofil og undernummer er en formidabel forbedring. Tidligere kunne en holde på med slike arbeidsoppgaver i dager og uker uten at en følte noen fremdrift og en gikk fort lei. Nå er det nesten som en gleder seg til en slik jobb.”*

En annen bruker sier at produktet forenkler hverdagen og gjør at man slipper en rekke unødvendige tastetrykk som man før måtte gjennom. Han mener dette reduserer nakke- og ryggskader i hele organisasjonen.

## 4.2 BibTid

BibTid er en webapplikasjon som tilbyr tidsskriftsformidling av norske og utenlandske tidsskrifter til biblioteker rundt omkring i Norge. I stedet for at bibliotekene kontakter utgiver direkte, går bestilling av abonnement fra forskjellige utgivere via denne tidsskriftsformidlingen. Tidligere har opprettelse og vedlikehold av abonnemeter foregått manuelt via telefon eller telefaks, dvs. at bibliotekarene har søkt etter tidsskrifter andre steder, og ringt bestillingene inn. Målsetningen med den nye webapplikasjonen er at både søking etter tidsskrifter og opprettelse, oppsigelse og vedlikehold av kundeforhold skal kunne skje også via web som en utvidelse av den eksisterende tjenesten.

Applikasjonen ble laget av tre utviklere i samarbeid med to brukere. En oversikt over hovedtrekkene av erfaringene fra BibTid er vist i tabell 4.3 på neste side. Brukerne var bibliotekarer ansatt i kundeorganisasjonen. Brukerne var i følge utviklerne gode brukerrepresentanter siden de har samme utdanning og erfaringsbakgrunn som nåværende brukere. Kundeorganisasjonen skal også benytte deler av applikasjonen i sitt daglige arbeid, og bør dermed kunne regnes som egentlige brukere. Initielt ble kravspesifikasjonen lagd av de to brukerne i samarbeid med et annet konsulentfirma i flere år gjennom en evolusjonær tilnærming. Opprinnelig var tanken at konsulentfirmaet skulle utvikle hele applikasjonen, men da de overskred tidsskjemaet med over 100%, resulterte dette i at kunden bestemte seg for å skifte leverandør. Kravspesifikasjonen hadde da vært gjennom flere iterasjoner, og hadde blitt tenkt vel igjennom før den ble overlevert til utviklerne på Genera. Webapplikasjonen ble etter dette utviklet av Genera i samarbeid med brukerne. Brukerne deltok på tre møter underveis i utviklingen; et møte initielt og to møter mens utviklingen pågikk. Her viste man fram produktet så langt som det var kommet, og brukerne kom med sine reaksjoner. Produktutviklingen foregikk da relativt smertefritt uten store endringer på arkitekturnivå, antagelig som et resultat av modne krav som hadde vært gjennom

<b>Aktivitet</b>	<b>Hovedpunkter</b>	<b>Beskrevet under</b>
<b>Estimering</b>	Ingen estimatsoverskridelser	4.2.1 på neste side
<b>Prosessmodell</b>	Den evolusjonære Genovaprosessen	4.2.2 på neste side
<b>Kravspesifikasjon</b>	Hadde vært gjennom mange iterasjoner over flere år da et annet firma hadde ansvar for prosjektet. Trengte dermed ingen gjennomgripende forandringer da Genera overtok prosjektet.	4.2.3 på neste side
<b>Prototyping</b>	Bestod av statiske HTML-sider som viste utlegget. Dette ble presentert for kunden, som evaluerte forslaget. Endringer ble gjort fortløpende basert på kundens endringsforslag. Etter revideringer ble produktet forevist kunden på nytt.	4.2.4 på side 52
<b>Endringshåndtering</b>	Kunden/brukerne kom ikke med noen radikale endringsønskter. Derimot var det en del lange diskusjoner rundt småendringer underveis i utviklingen. Utviklerne synes alle diskusjonene og småendingene tok for lang tid, og at dette burde vært håndtert på en mer formell måte.	4.2.5 på side 52
<b>Brukermedvirkning</b>	Utviklerne syntes brukerne brukte for lang tid på avgjørelser, og på å gjøre om tidligere valg. Brukerne mener deres rolle i prosjektet var meningsfull og bidro til å gjøre produktet bedre tilpasset deres egentlige ønsker for produktet. Men de beskriver det som vanskelig å se for seg hvordan det ferdige produktet ville se ut idet de foretok valg.	4.2.6 på side 52

Tabell 4.3: Oversikt over erfaringer fra prosjektet BibTid

flere iterasjoner. Men uansett var det en ganske stor mengde av kontinuerlige endringsønsker fra brukerne rundt detaljer på produktet. Endringsønsker ble også sendt inn på deler av produktet som allerede hadde vært endret flere ganger, fordi brukerne syntes det var vanskelig å forutsi konsekvensene av endringsønskene de kom med. Utviklerne klaget på at for mye tid ble kastet bort på diskusjoner med brukerne rundt detaljer ved endringsønskene. Brukerne syntes på sin side at disse diskusjonene var fruktbare, og syntes at produktet ikke ville ha blitt tilpasset deres endelige behov, hvis de ikke hadde hatt disse diskusjonene.

Webutviklingsprosjektet var et fastprisprosjekt. Som prosessmodell brukte utviklerne den evolusjonære Genovaprosessen. Prosjektet hadde en varighet på seks måneder, og opplevde ingen estimatoverskridelser. Som sluttprodukt leverte man en kjørbar leveranse. Etter leveranse hadde kunden en godkjenningssperiode der de kunne forsikre seg om at produktet ikke avvek fra kravspesifikasjonen.

#### **4.2.1 Estimer**

På tross av alle endringsønskene fra brukerne, ble ikke kostnadsestimatene oversteget på BibTid.

#### **4.2.2 Prosessmodell**

På BibTid fulgte man i store trekk den evolusjonære Genovaprosessen. Design, analyse og kode ble gjort iterativt. De mest kritiske delene ved systemet ble testet ut først, dvs ytelsestid og responstid. Man opplevde etter dette ingen store overraskelser underveis.

På de tre møtene med kunden viste utviklerne fram produktet så langt som det var kommet, og kunden kom med sine reaksjoner på det produktet som da forelå. Man kan beskrive hvert møte med kunden som en del-leveranse som da ble evaluert.

#### **4.2.3 Kravspesifikasjon**

Selv om både kravspesifikasjon og designrapport lå klar for utviklerne hos Genera ved prosjektoppstart, var det mange detaljer som måtte bli avklart på møter og på e-post underveis i utviklingen. Kravspesifikasjonen gjennomgikk ingen radikale endringer underveis i den delen av utviklingen som Genera deltok i, og dokumentet står uforandret ved prosjektets ferdigstilling. Brukene kom ikke med noen radikale endringsønsker underveis i prosjektforløpet — endringsønsker var på detaljnivå.

#### 4.2.4 Prototyping

Man utviklet ingen prototype i tradisjonell forstand på produktet, men man hadde heller ingen ukontrollert utvikling som det var umulig å gå tilbake på. Først lagde man tilbudet, som bestod at statiske HTML-sider som viste utlegget. Ved oppstart lagde en av utviklerne så alle websidene applikasjonen skulle bestå av, uten å legge noen avansert logikk eller funksjonalitet bak. Dette var tenkt som basis for diskusjoner rundt kravspesifikasjon, og var slik sett en prototypertilnærming. Dette produktet ble presentert for kunden, som så evaluerte forslaget. Endringer ble etter møtene gjort fortløpende basert på kundens endringsforslag. Etter revideringer og videreutvikling av websidene med underliggende funksjonalitet, ble produktet vist for kunden på nytt. Kunden hadde ikke tilgang til produktet gjennom utviklingsfasen, men virket fornøyd med måten dette ble gjort på.

#### 4.2.5 Endringshåndtering

På BibTid kom ikke kunden/brukerne med noen radikale endringsønsker underveis i prosjektforløpet. Derimot kom de med en del presiseringer og hadde en del diskusjoner rundt detaljer underveis i utviklingen. Detaljeringene og presiseringene ble forsøkt nedfestet i form av møtereferater og e-postkorrespondanse, og ble sett på som bindende hvis man hadde skriftlig dokumentasjon på dem. Leverandør var ikke fornøyd med endringshåndteringen hos kunde. Samtlige utviklerne synes alle småendringene kunden ønsket seg tok for lang tid. Mye tid ble brukt på å rette opp skrivefeil, bestemme ordlyden på menyer og på å standardisere begreper. Utviklerne mener at de etter dette har lært å dokumentere alle diskusjoner med kunde. Den ene utvikleren opplevde det som frustrerende at kunden etter endt diskusjon kunne starte den samme diskusjonen på nytt etter et par uker, og denne gangen komme opp med et annet resultat. Dette fikk de ikke tilbakevist overfor kunden i de tilfellene der de ikke hadde dokumentert utfallet av diskusjonene.

#### 4.2.6 Brukermedvirkning

Utviklerne syntes kunden hadde vanskelig for å bestemme seg. På møtene med kunden oppstod en del interne stridigheter rundt presiseringer og detaljer, og utviklerne mente disse burde vært "tatt på kammerset". Et typisk tilfelle var at teksten som skulle stå på en av knappene i applikasjonen ble diskutert fram og tilbake i en time foran utviklerne. Dette fortonte seg som langtekkelige diskusjoner uten annen relevans for utviklerne enn at de ville vite hva kunden til slutt bestemte seg for. Utviklerne tror at mye av famlingen rundt spesifikasjoner og detaljering skyldtes at kunden var uerfaren med å delta i utviklingsprosjekter.



Brukerne beskriver utviklerne som fleksible og imøtekommende, og opptatt av at produktet skulle bli bra. De synes utviklerne var lydhøre og runde i kantene når det gjaldt å håndtere presiseringene de ønsket seg. De hadde også inntrykk av at utviklerne forstod problemene de hadde, og brukerne syntes samarbeidet fungerte knirkefritt hele veien. Derimot var det slitsomt å være bruker i utviklingsprosessen, fordi deltakelse i prosjektarbeidet kom i tillegg til ordinære arbeidsoppgaver.

Brukerne mener at ved å delta i utviklingen fikk man *“retta på noen ting som buttet litt, og fikk gjort noen justeringer som gjorde produktet enklere å bruke”*. Brukerne mener deres rolle i prosjektet var meningsfull og bidro til å gjøre produktet bedre tilpasset deres egentlige ønsker for produktet. De mente videre at grunnen til småendringene og presiseringene underveis var at de kunne bestemme seg for en ting, men da de så konsekvensene av bestemmelsene var det ikke alltid slik de ønsket seg det likevel. De beskriver det som vanskelig å se for seg hvordan det ferdige produktet ville se ut idet de tok valg. Her ser man med andre ord et gap mellom utviklerne og brukerne når det gjelder erfaringshorisont og opplevelse av brukermedvirkning.

## 4.3 UniBase

UniBase er et administrativt system for virksomhetsoppfølging på svenske universiteter og høyskoler. UniBase-databasen inneholder bearbejdede data fra forskjellige administrative systemer fra en del svenske høyskoler og universiteter. Informasjonen som finnes i databasen, prosesseres av rapportbestillingsapplikasjonen som nå har et webgrensesnitt. Rapportbestillingen henter data fra de ønskede databasene og genererer disse i en fil ut i fra hvilke parametere brukeren ønsker å ha med. Rapportfilen som er generert kan leses i et regneark (f.eks. MS Excel dersom man har Windows OS). En oversikt over hovedtrekkene av erfaringene fra UniBase er vist i tabell 4.4 på neste side.

### 4.3.1 Estimering

Utviklingen av applikasjonen tok fem måneder, og prosjektet opplevde ingen estimatoverskridelser. Prosjektet var timebasert. Det var fire utviklere på prosjektet. I starten av prosjektet utarbeidet leverandør et prosjektilbud der kunden kunne velge mellom en enkel og en mer avansert løsning. Kunden foretrakk da den enkleste. Basert på dette valget ble det utarbeidet et mer detaljert løsningsforslag som ble presentert for kunden på et møte. Brukerne ønsket ikke å delta i utviklingen av webapplikasjonen.

<b>Aktivitet</b>	<b>Hovedpunkter</b>	<b>Beskrevet under</b>
<b>Estimering</b>	Ingen estimatoverskridelser	4.3.1 på forrige side
<b>Prosessmodell</b>	Evolusjonær, men ingen stramt definert prosessmodell	4.3.2 på neste side
<b>Kravspesifikasjon</b>	Siden formålet var å gjøre et allerede eksisterende system med tradisjonelt grensesnitt om til en webapplikasjon, ble det eksisterende klient-tjener-systemet gitt som kravspesifikasjon sammen med diverse føringer	4.3.3 på neste side
<b>Prototyping</b>	Ingen prototyping	
<b>Endringshåndtering</b>	Det kom ikke noen endringsønsker underveis, verken fra kunde/brukere eller internt i utviklerorganisasjonen	4.3.4 på neste side
<b>Brukermedvirkning</b>	Kunden ønsket ikke deltagelse fra brukere i utviklingen	4.3.5 på neste side

Tabell 4.4: Oversikt over erfaringer fra prosjektet UniBase

### 4.3.2 Prosessmodell

UniBase-prosjektet var evolusjonært, men fulgte ingen stramt definert prosessmodell. Dette skyldtes delvis den sparsomme kontakten med kunden, men også at “kravspesifikasjonen”, dvs systemet som skulle gjøres om til en webapplikasjon, var velkjent og uten store usikkerhetsmomenter. Videre mente prosjektlederen at behovet for en prosessmodell var mindre enn vanlig fordi prosjektet var lite i både størrelse og varighet.

Flytdiagrammer ble tegnet på et veldig enkelt nivå. Til å begynne med gjorde prosjektlederen mye administrativt arbeid, som å tegne detaljerte sekvensdiagrammer og legge ut mye informasjon på web, men fant ut at dette var for tidkrevende i forhold til nytten de fikk ut av det. Prosjektdeltakerne fant ut at det var lettere å kommunisere direkte. Kostnadsestimatene ble ikke oversteget, og produktet ble levert på avtalt tidspunkt.

### 4.3.3 Kravspesifikasjon

Formålet med UniBase var å gjøre et allerede eksisterende system som hadde et tradisjonelt grensesnitt, om til en webapplikasjon. Som utgangspunkt hadde konsulentene det eksisterende klient-tjener-systemet. Det fantes dermed ingen skriftlig kravspesifikasjon for prosjektet, men det gamle systemet fungerte som en kravspesifikasjon sammen med diverse føringer som var lagt fra kunden. Grunnen til at kunden ønsket å gjøre den tradisjonelle applikasjonen om til en webapplikasjon, var at de ønsket å gjøre den mer tilgjengelig. Å ha systemet på et intranett, impliserer at man slipper å installere det på hver enkelt PC. Det var ingen endringer på denne “kravspesifikasjonen” underveis i utviklingen, men det kom heller ingen endringsønsker fra brukerne underveis i utviklingen.

### 4.3.4 Endringshåndtering

På UniBase opplevde man ingen problemer relatert til endringshåndtering. Brukerne/kunden kom ikke med noen endringsønsker underveis, og det oppstod heller ingen store problemer med produktet underveis i utviklingen som medførte endringsønsker internt i utviklerorganisasjonen.

### 4.3.5 Brukermedvirkning

Kunden virket uinteressert i å delta i utviklingen, og skaffet heller ingen brukere til evalueringsaktiviteter. Prosjektdeltakerne fra utviklerorganisasjonen hadde ønsket seg noe mer kontakt med kunden.

Da produktet ble levert kunden for installasjon, hørte utviklerne ikke ett ord fra kunden før det hadde gått to måneder. Da kom det en forespørsel om hvordan de skulle installere produktet. Siden det hadde gått såpass lang

tid, var utviklerne da allerede allokert til andre utviklingsprosjekter, og siden kontraksperioden var over, måtte all assistanse og feilretting nå betales av kunden.

## 4.4 Retningslinjer for prosess

Som en konklusjon for erfaringsinnhenting i de tre prosjektene er de følgende retningslinjene for prosess laget. Retningslinjene er generelle retningslinjer ment som innspill til Genova webprosess. Siden erfaringsinnhenting beskrevet ovenfor viste at endringshåndtering var et særlig problemfylt område, vil endringshåndtering bli behandlet mer detaljert i neste kapittel.

### 4.4.1 Prosjektetablering og planlegging

*Ta høyde for kompetanseoppbygging og bruk av umoden teknologi i estimering av prosjektkostnader for webutviklingsprosjekter. Vær oppmerksom på at endringer kan forekomme fordi teknologien ikke fungerer slik man hadde tenkt seg det.*

- Erfaringer fra alle de tre prosjektene viste at det å lage webapplikasjoner krevde en god del kompetanseoppbygging og tilegning av mye ny teknisk kunnskap. Dette i kombinasjon med bruk av umoden teknologi kan medføre endringer fordi teknologien fungerer på en annen måte enn man hadde tenkt seg, eller at det ikke fungerer på områder man hadde tenkt seg at den ville fungere. Dette må man ta hensyn til når man skal estimere de totale prosjektkostnadene ved prosjektetablering. Teknologien som ble benyttet i webutviklingsprosjektene opplevdes som umoden, tidkrevende og full av feil. Det kan virke som om webteknologi er vanskeligere å bruke enn teknologi for mer tradisjonell utvikling, fordi den er mindre utprøvd, mer umoden og dermed inneholder mer feil.
- Vær sikker på at den antatte teknologien for funksjonelle og ikke-funksjonelle krav for applikasjonen støttes av de valgte webutviklingsverktøyene.
- Videre var det på samtlige prosjekter synspunkter på at viktig funksjonalitet på web begrenser seg sterkt eller er helt manglende i forhold til hvordan applikasjonen kunne blitt dersom man brukte tradisjonelle (det vil si mer operativsystemavhengige) utviklingsverktøy. Manglende funksjonalitet medfører dermed at brukervennlighet er vanskeligere å få til på web. Dette skyldes at verktøyene ikke er så modne, og at man dessuten er bundet av hva de ulike browserne støtter. Dermed bør det aktuelle utviklingsverktøyet bli undersøkt nøye for å forsikre

om at det gir adekvat støtte for den tiltenkte funksjonaliteten og for de ikke-funksjonelle systemkravene.

#### 4.4.2 Brukermedvirkning

*Tilstreb å komme i kontakt med brukerne – og helst så tidlig som mulig i utviklingsprosessen. Det kan være svært kostnadsbesparende dersom brukerne er involvert fra starten av prosjektet. Brukerne kan da bidra til å evaluere kravspesifikasjonen. De kan dermed bidra til å eliminere unødvendig implementasjon før den blir implementert, til å redusere for kompleks funksjonalitet og til å gjøre endringer mens disse ikke er så kostbare som på et senere stadium.*

- På TelMont kom brukerne inn i utviklingen da den hadde pågått en stund, og dette medførte mange ekstra endringer fordi brukerne hadde en helt annen oppfatning av hvordan systemet burde være enn de kunderepresentantene som hadde formulert kravspesifikasjonen. Alle parter på TelMont var enig i at dette ville vært kosteffektivt å involvere brukerne på et atskillig tidligere stadium, fordi det da ville kommet færre endringsforslag.
- Å redusere kompleks funksjonalitet medfører ikke bare at man reduserer kostnader, men også at man tilpasser produktet bedre mot hva brukerne egentlig trenger. I et tilfelle gjorde det mindre kompliserte produktet sitt til at det ble enklere både å bruke og vedlikeholde.
- Tidlig brukermedvirkning kan bidra til at man eliminerer unødvendig funksjonalitet før den implementeres. I to av prosjektene (TelMont og UniBase) så man at brukerne ønsket seg et enklest mulig produkt. Kunnskap om dette på et tidligere stadium i TelMont kunne bidratt til å eliminere mye implementasjonsarbeid før det var påbegynt.
- I de to prosjektene med brukermedvirkning, synes både kunde- og utviklerorganisasjon at produktet hadde høyere kvalitet ved levering som et direkte resultat av brukermedvirkningen. De leverte begge riktige produkter – produkter som ble applaudert av brukerne. Alle endringsforslagene fra brukerne hadde bidratt til at de fikk et produkt bedre tilpasset deres egentlige behov. Da brukerne fikk sett produktet underveis i utviklingen, forstod de at kravspesifikasjonen inneholdt mangler eller ikke var klart nok formulert, og foretok så de nødvendige justeringer ved å sende inn endringsforslag til utviklerne.
- Alle brukerne syntes at deres deltagelse i prosjektet var nyttig, og resulterte i et produkt som var bedre tilpasset deres arbeidsprosesser. Brukerne syntes at deres tilstedeværelse i prosjektet var nødvendig for å sikre implementasjon av viktig funksjonalitet. Brukerengasjementet

bidro også til at brukerne var mer positivt innstilt til å bruke produktet etter levering, og til å lære bort produktet til andre brukere.

- Brukermedvirkning kan resultere i mer fornøyde brukere. Brukerne var mer fornøyde med prosjektet de hadde vært involvert i enn i andre lignende prosjekter der man ikke hadde benyttet seg av brukermedvirkning. Brukerne var fornøyde delvis fordi de hadde hatt muligheten til å si ifra om sine synspunkter og komme med endringsønsker, samtidig som de kunne avfeie funksjonalitet som de så på som unødvendig. De synes også selv de ga nødvendige innspill på glemte funksjonalitet og arbeidsprosesser som var nødvendige for å lage et velfungerende produkt. Dette taler for at det er vel verdt å implementere endringsforslag som kommer fra brukerne, på tross av at endringene kan bli mange og ta mye tid. For det viktigste er jo tross alt å lage et riktig produkt.

*Ikke forvent umiddelbar feedback fra brukerne. Gi dem tid til å bli kjent med de forskjellige (inkrementelle) produktleveransene. Det er først etter en stund de klarer å komme med fornuftige endringsønsker.*

- Brukerne kom med endringsønsker først etter at utviklingen hadde pågått en stund, det vil si da de hadde blitt mer kjent med produktet og dermed bedre kunne se muligheter for forbedring enn i begynnelsen. Alle brukerne beskrev det som vanskelig å se for seg løsningen basert på tekniske spesifikasjoner, men mente at det ble atskillig lettere straks de fikk noen prototyper å visualisere ut i fra.
- Utviklerorganisasjonen bør være klar over at brukerne i de fleste tilfeller allerede har en full arbeidsdag fylt med deres daglige plikter, og at å komme med endringsforslag er en ekstra arbeidspost for dem. Derfor burde utviklerorganisasjonen megle med ledelsen til brukerne for å hjelpe dem å bli fristilt fra noen av deres daglige gjøremål i en periode dersom de ønsker seg endringsforslag.

#### 4.4.3 Prototyping

*Fokuser tidlig prototyping på en teknologievaluering for å teste at teknologien er som forventet. Atskill klart teknologiprototypingen fra prototypingen av kravspesifikasjonen slik at ikke mange endringer gjøres på basis av feil valgt teknologi.*

- I et av prosjektene ble for mye arbeid lagt i den kombinerte teknologi/kravspesifikasjonsprototypen. På det tidspunktet da prototypen var kjørbar, var den blitt svært stor. Fordi det var lagt så mye arbeid ned i prototypen, ble det bestemt at man skulle basere produktet på denne prototypen. Denne beslutningen ble tatt fordi det var økonomisk umulig å forsvare hvorfor den valgte teknologien skulle skiftes ut, på

tross av at prototypen viste at den hadde store integrasjonsproblemer. Teknologien viste seg også å ha betydelige feil og hadde sine klare begrensninger. Det anbefales derfor å lage en enkel teknologiprototype i starten av prosjektet for å teste den valgte teknologien, og å separere denne prototypen fra annen prototyping. Å teste ut teknologien før man spør etter endringsforslag og starter med en prototyping av kravspesifikasjonen er også å anbefale, slik at man ikke foretar mange endringer på basis av en prototype som likevel skal skiftes ut fordi det viser seg at teknologien ikke holder.

- Brukere er høyt kvalifiserte for å evaluere produktfunksjonaliteten, men de har ikke nødvendigvis teknisk kompetanse, og kan ha problemer med å forstå tekniske detaljer ved produktet før disse blir vist i praktisk bruk. Dette gapet i teknisk kunnskap mellom utviklere og brukere kan forhindre effektiv kommunikasjon. Bruk prototyping som en måte å minske dette gapet mellom utviklernes og brukernes tekniske kunnskap og ordforråd.
- Brukerne i to av prosjektene rapporterte om problemer med å forutse konsekvensene av valgene deres, særlig når det kom til tekniske detaljer. De kunne derfor ombestemme seg da endringsønsket deres var implementert, og de skulle evaluere resultatet av endringsønsket. Dette opplevdes som irriterende og frustrerende for utviklerne. Utviklere bør være klar over at dette kan skje, og for komplekse endringsforespørsler kan det være tilrådelig å lage en liten prototype som brukeren kan evaluere før man putter for mye energi inn i utviklingen av endringsønsket.
- Brukerne i to av prosjektene beskrev det som vanskelig å forestille seg hvordan produktet ville komme til ut da de bare leste kravspesifikasjonen. De beskrev det som lettere å evaluere kravspesifikasjonen når utviklingen hadde pågått en stund, slik at de så noen deler av det ferdige produktet og i større grad skjønnte hvordan produktet som helhet ville bli. Fossefallsmodellen åpner ikke for de samme mulighetene for at brukerne kan evaluere produktet mens det er under utvikling, og det kan derfor være vanskelig for dem å avgjøre hvorvidt de er enige i kravspesifikasjonen eller ikke. En evolusjonær utviklingsmetodikk tillater brukerne i større grad å skifte mening underveis i utviklingen. Det er dermed lettere å lage produktet tettere knyttet opp mot brukernes egentlige behov.
- En prosjektleder hos en kunde beskriver prototypene som god hjelp i å kartlegge krav og å skape et grunnlag for det som skulle utvikles. Prosjektlederen tror også det er nødvendig å lage og utvikle på en prototype for å kunne presentere endringene for brukerne. Prototypen viste fram konkrete endringer, og ikke bare endringsforslag på papir.

Det å kunne vise fram en prototype var for brukerne en fin måte å få brukerne til å visualisere produktet, framfor å bare bruke masse fagterminologi og prosa som ikke ville synliggjort produktet for dem i samme grad.

#### 4.4.4 Struktur rundt endringshåndtering

*Bedre kommunikasjon rundt endringshåndtering i utviklingsprosjekter kan øke den gjensidige enigheten rundt reviderte kostnader og tidsplaner mellom de ulike aktørene i et utviklingsprosjekt. Det anbefales at kunde, brukere og utviklerorganisasjon blir enige om og følger en formell prosess for endringshåndtering. For å sikre at dette gjøres konsistent, anbefales det å bruke et verktøy for endringshåndtering. Verktøyet kan støtte kommunikasjonen mellom kunde, utviklere og brukere. Et eksempel på et slikt verktøy vil bli beskrevet senere i oppgaven.*

- I evolusjonær utvikling kan det lett oppstå problemer rundt kostnadskontroll som et resultat av nye endringsforespørsler fra brukerne underveis i utviklingen. Når brukerne foreslår nye endringer som bedre tilpasser produktet til deres egentlige behov, er det ikke nødvendigvis slik at dette bare blir sett på som fordelaktig fordi dette kan gå på bekostning av kontrollen over kostnader og tidsskjemaer.
- To prosjekter opplevde alvorlige kommunikasjonsproblemer og mye frustrasjon som et resultat av at de ikke var enige i en formell endringshåndteringsprosess. Alle prosjektmedlemmene rapporterte et behov for en formalisert prosess som kunne holde oversikt over alle endringsønskene. Utviklerne i et prosjekt var frustrerte fordi kunden og brukerne stadig ombestemte seg, og mange deler av produktet måtte bli endret flere ganger. I det andre prosjektet var kunden veldig misfornøyd over ikke å bli konsultert om hvorvidt visse endringsønsker var aksepterbare fra et økonomisk perspektiv. Begge disse problemene kunne vært unngått dersom man hadde hatt en mer strukturert prosess rundt endringshåndtering. Prosessen burde formalisere kommunikasjonen mellom kunde, utviklere og brukere slik at endringshåndtering og kostnader kommer mer under kontroll.
- I et av de timebaserte prosjektene var kunden veldig frustrert over de hyppige endringene i kravspesifikasjonen, fordi disse endringene forårsaket at prosjektet ble forsinket og at det ble atskillig dyrere enn planlagt. De kontinuerlige endringene i kravspesifikasjonen ble på dette prosjektet en kilde til mye diskusjon. Prosjektleder hos kunde følte at han mistet kontroll fordi kravspesifikasjonen endret seg kontinuerlig. Men utviklere trodde sterkt på at endringsønskene fra brukerne var nødvendige for å tilpasse produktet til deres egentlige behov. Resultatet ble



en stor diskusjon rundt økonomi. En mer formell endringshåndteringsprosess kunne ha unngått noen av disse erfarte problemene.

#### 4.4.5 Avsluttende bemerkninger

Å fokusere på endringshåndtering i kommende evolusjonære webutviklingsprosjekter kan føre til:

- At man er klar over at endringer som gjøres underveis, kan føre til at brukerne blir mer fornøyde med det endelige produktet, både fordi de har fått delta i bestemmelser, men også fordi endringene kan tilpasse produktet mer etter deres *egentlige* behov.
- At man beregner endringshåndtering inn i estimatene slik at man ikke får seg en stor overraskelse når endringene begynner å strømme inn, og man ikke har beregnet tid til disse.
- At man er bevisst på å involvere brukerne så tidlig som mulig, slik at unødvendig funksjonalitet kan elimineres på et så tidlig stadium som mulig.
- At man får en produktutvikling som tar hensyn til alle parter under bestemmelsen rundt hvorvidt endringsønskene bør implementeres eller ikke.
- At man strever etter å få ledelsen til brukerne til å fristille dem fra noen av deres daglige gjøremål en periode slik at de kan sette av tiden til å komme med endringsønsker.
- At brukerne underveis i utviklingen spør seg om viktig funksjonalitet som er glemt, og er trygge på at de kan komme med endringsønsker som inkluderer denne.
- At utviklerne er klar over gapet i erfaring mellom utviklere og brukere som oppstår rundt det tekniske ordforrådet, og at utviklerne er klar over at dette gapet nødvendigvis må medføre endringsønsker.
- At bedre kommunikasjon rundt endringshåndtering kan øke den gjensidige enigheten rundt reviderte kostnader og tidsplaner.

## 4.5 Relatert arbeid

Resultatene presentert i dette kapitlet bekrefter i stor grad resultater fra relaterte studier, selv om det også er noen forskjeller. Erfaringene vil nå sammenlignes med funn fra ulike studier.

### 4.5.1 Emam et. al

Et studie foretatt av Emam samsvarer svært godt med resultatene i denne oppgaven (Emam, Quintin & Madhavji 1996). Emam et. al mener at brukermedvirkning er fordelaktig i systemutvikling der kravene er usikre. Men når usikkerheten rundt kravene er liten, minskes fordelene ved brukermedvirkning. På slike prosjekter kan brukermedvirkning tvert i mot fortone seg som irriterende for brukerne fordi de føler at de ikke kan bidra med noe. Denne motstanden kan videre føre til en forringelse av kravspesifikasjonen og kvaliteten på produktet. Disse resultatene er i samsvar med observasjonene av den vellykkede brukermedvirkningen i TelMont og BibTid versus den manglende brukermedvirkningen UniBase opplevde. På UniBase-prosjektet var brukerne ikke villige til å delta i utviklingen. Fra deres synspunkt var kravspesifikasjonen gitt, det vil si at de ønsket en tradisjonell applikasjon over på web, og dette innebar for dem ikke noen usikkerhetsmomenter. De virket fornøyde med det ferdige produktet på tross av deres sparsomme kontakt med utviklerne gjennom utviklingsprosessen. Motsatt var det i de to andre prosjektene, der brukerne ikke visste eksakt hvilke krav eller hva slags funksjonalitet det ferdige produktet burde inneholde. Her ble kravspesifikasjonen forbedret og endret flere ganger gjennom prototyping og dialoger med utviklerne. Sluttbrukerne i disse prosjektene var svært samstemte på at de syns deres deltagelse i utviklingen var viktig og nyttig for å utvikle et *riktig* produkt som var tilpasset deres behov. Dette skyldtes at de ikke klarte å se hva de egentlig ønsket seg før de så prototypene. Deres deltagelse minsket også utviklernes usikkerhet rundt kravspesifikasjonen.

### 4.5.2 Saarinen et. al

Saarinen et. al gir gjennom empiriske studier belegg for at brukermedvirkning er viktig, men at det ikke finnes noen “magisk formel” som gjør at et prosjekt automatisk vil lykkes dersom brukerne inkluderes i prosjektutviklingen (Saarinen & Saaksjarvi 1990). Saarinen et. al mener derimot at utviklernes evne til å samarbeide og kommunisere med sluttbrukerne er sentralt, slik at kvaliteten på brukermedvirkningen er det avgjørende faktoren for om et prosjekt lykkes eller ikke. Empiriske funn foretatt i studiet viser at det ikke er kvantiteten på brukermedvirkningen som er avgjørende for hvorvidt prosjektet vil lykkes. Derimot spiller dyktige systemutviklere med gode kommunikasjonssevner inn på hvorvidt prosjektet vil lykkes eller ikke, uavhengig av

i hvor stor grad brukerne har vært deltakende i prosjektutviklingen. Relatert til funnene i denne oppgaven, der produktene var en suksess for brukerne på både TelMont og BibTid, fungerte samarbeidet tilfredsstillende i de to prosjektene. Muligens kan dette gode samarbeidet med sluttbrukerne ha innvirket på prosjektet slik at det ble en suksess.

### 4.5.3 Zamperoni et. al

Prototyper kan være en god måte å kommunisere med brukerne på (Grudin 1991, Floyd 1984, Ehn 1993, Bratteteig 1997, Bansler 1989, Zamperoni et al. 1995, Mathiassen 1995, Lien et al. 2001, Wiig 1994, Sefland 1998). Et studium foretatt av Zamperoni et. al viser at brukere eller kunder noen ganger har en spesiell visjon av det kommende systemet, men at de ikke klarer å formulere denne visjonen til utviklerne (Zamperoni et al. 1995). Zamperoni et. al mener dette problemet kan forminskes gjennom en kontrollert evolusjon av systemet gjennom prototyping og hyppig brukerrespons. Prototypene er et nyttig hjelpemiddel for kommunikasjon – de blir en formidler, en slags mekler for kommunikasjon, slik at utviklerorganisasjonen kan komme til enighet med brukerne om hvordan systemet bør bli og hvilke forventninger brukerne kan få innfridd. I samsvar med våre resultater, viste studiet at sannsynligheten for at prosjektet blir akseptert øker dersom sluttbrukerne involveres i utviklingsprosessen. Zamperoni et. al understreker at dette forutsetter at det er korte kommunikasjonskanaler mellom utviklere og sluttbrukere, noe som var tilfelle både på TelMont og BibTid. Lignende resultater har også blitt rapportert av Ehn (Ehn 1993). Han argumenterer for at sluttbrukere er høyt kvalifiserte for å evaluere produktfunksjonaliteten, men at de ikke nødvendigvis har de tekniske kunnskaper for å skjønne kravene før de blir demonstrert i praktisk bruk. Ehn poengterer at dette gapet i tekniske kunnskaper kan bli jevnet ut dersom man benytter seg av prototyper. Samme observasjon ble gjort i vårt studium, der brukerne fortalte at det var lettere å bidra etter en stund – når de var i stand til å se noen deler av det endelige produktet, ikke bare en teknisk spesifisering.

### 4.5.4 Floyd

Floyd (Floyd 1984) mener prototyper bringer feedback og kommunikasjon inn i utviklingen, og kan bidra til at man utvikler mer adekvate systemer. Hun mener en kombinasjon av analytiske og eksperimentelle aktiviteter er fruktbart, noe som stemmer overens med erfaringene fra TelMont-prosjektet. En slik kombinasjon kan blant annet bestå av at man analyserer situasjonen til brukerne, lager en kravspesifisering, lager en prototype for å få den evaluert av brukerne og utvikler målsystemet inkrementelt. Dette er mønstret som ble fulgt i TelMont-systemet. Floyd mener denne måten å utvikle på gjør at man lett får kontroll over utviklingsprosessen. Denne antagelsen

stemmer ikke udiskutabelt godt overens med det man opplevde på TelMont, der prosjektlederne opplevde frustrasjon og mangel på kontroll. Derimot er det rimelig å anta at mangelen på kontroll ikke skyldtes utviklingsmetoden, men mer en mangel på struktur over kommunikasjonen mellom kunde, brukere og leverandør rundt endringshåndteringen. Floyd mener videre at bruken av prototyper og diskusjoner rundt avgjørelser med brukerne under prototypingen, gjør at brukerne føler en sterkere forpliktelse overfor det ferdige produktet. Dette stemmer godt overens med våre resultater, som viste at brukerne følte en forpliktelse for å følge opp produktet når det ble innført i organisasjonen deres.

#### 4.5.5 Mathiassen

I likhet med Floyd, foreslår Mathiassen at man kombinerer analytiske og eksperimentelle aktiviteter (Mathiassen 1995). Mathiassens empiriske studier av ulike tilnærminger til programvareutvikling viser at en blandet evolusjonær tilnærming, der man både gjør spesifisering og prototyping under utviklingen, gir svært gode resultater for sluttproduktet med hensyn på funksjonalitet, robusthet, brukbarhet og hvor lett produktet blir å lære. Dette skyldes at den blandede tilnærmingen tilsynelatende kombinerer de sterke sidene til de to spesialiserte tilnærmingene, og kombinerer analytiske og eksperimentelle tilnærminger til programvareutvikling på en heldig måte. Man fant også ut at spiralmodellen er et nyttig rammeverk. Dette samsvarer svært godt med det gode resultatet man fikk på TelMont-prosjektet, der man både spesifiserte og prototypet gjennom en evolusjonær tilnærming.

#### 4.5.6 Lichter et. al

Lichter et al. har derimot et kritisk syn på brukermedvirkning (Lichter et al. 1994). Resultatene her er basert på intervjuer med utviklere. Studiet viste at utviklerne kan gjøre den feilen at de oppmuntrer brukerne til å si ifra om alle mulige ideer som kommer til dem når de evaluerer en prototype. Dette kan medføre at man inkluderer enhver funksjon eller designmulighet i produktet, med det resultat at brukbarheten minsker og kompleksiteten på koden øker. Dette var *ikke* tilfelle i vårt studium. Spesielt i TelMont-prosjektet reduserte brukermedvirkning i stedet kompleksiteten på kravspesifikasjonen og det ferdige produktet.

#### 4.5.7 Bratteteig

På TelMont-prosjektet resulterte brukermedvirkningen i en mindre komplisert kravspesifikasjon og gjorde produktet enklere både å bruke og vedlikeholde. Men som Bratteteig (Bratteteig 1997) understreker, er det ikke nødvendigvis en uproblematisk avgjørelse å ta for utviklerne å utvikle et enklest mulig produkt. Sluttbrukernes ønske om et produkt som er enklest

mulig å bruke kan kollidere med utviklernes ønske om å lage et produkt som tar i bruk ny teknologi og imponerende funksjonalitet som er utfordrende for dem å kode. For sluttbrukerne trenger derimot ikke teknisk eleganse å være viktig. På TelMont opplevde man denne spenningen. Ledelsen ønsket seg et produkt som inkorporerte mange tekniske detaljer og flere forskjellige veier å navigere seg gjennom systemet på. Sluttbrukerne ønsket på den annen side et system som var så enkelt som mulig, med én mulig vei å navigere seg gjennom systemet på. Å gi bort “makten” om designet på denne måten betyr at andre aspekter enn de tekniske stilles i fokus, noe som kan være uvant for teknisk orienterte utviklere (Bratteteig 1997). Det var en utfordring for utviklerne å akseptere dette, men de valgte etter hvert å gå for sluttbrukerens løsning. Resultatet var svært fornøyde brukere. Det samme var gjeldende i UniBase. I starten av utviklingen lagde utviklerne to prototyper som de viste kunden, og kunden foretrakk den enkleste. Utviklerne gikk med på dette, men fortalte at de selv hadde foretrukket å implementere den mest kompliserte løsningen hadde de kunnet bestemme selv.

#### 4.5.8 Bratteteig og Stolterman

Utviklingen av prototypene på TelMont ble utført i en gruppe, og resultatet av arbeidet som gruppa gjorde, var svært vellykket, med *“trampeklapp fra brukerne”* som prosjektleder hos kunde uttrykte det. Også utviklingen av BibTid-prosjektet ble utført i en gruppe bestående av utviklere og sluttbrukere. Også dette resultatet ble sett på som vellykket. Grupper kan i samarbeid få fram visjoner og bilder som ikke kan lages av et enkelt medlem av gruppa (Bratteteig & Stolterman 1997). I en gruppe kan forskjellige posisjoner og synsvinkler komme fram ved at man inkluderer ulike mennesker i gruppa. Forskjellige perspektiver er også viktige for dynamikken i en kreativ prosess. Dette forutsetter derimot at de ulike gruppemedlemmene er villige til å modifisere perspektivene sine. Har man ulike posisjoner innad i gruppa, tvinges de ulike medlemmene til å se på systemet fra forskjellige synsvinkler og dermed få et mer nyansert syn på det. Har man derfor en uensartet gruppe av utviklere og brukere, kan dette bidra til at man finner nye kvaliteter ved systemet som man ikke ville funnet ellers. Ut i fra erfaringene som er hentet inn, kan det virke som om denne dynamikken var fruktbar i både TelMont og BibTid-prosjektene.

#### 4.5.9 Ehn

Også Ehn argumenterer for at utvikling i en slik gruppe er positivt (Ehn 1993). Teknisk sett mener Ehn brukermedvirkning gir en blanding av teknisk kompetanse og brukernes opptatthet av utseende og brukervennlighet. Han tror dette kan bidra til et suksessfullt design og høykvalitetsprodukter. Som kontrast ser man ofte at “ekspertsystemer” utviklet av

bare ingeniører mislykkes i å få god brukbarhet på det endelige systemet, antagelig fordi ingeniørene tar enkelte faguttrykk og termer som gitt, og at operasjoner som for dem er selvsagte ofte ikke er like innlysende for en bruker som er mer uerfaren med datamaskiner. For å forklare hvordan systemutviklere og brukere kommuniserer, bruker Ehn grunnleggende elementer i Wittgensteins språkfilosofi (Wittgenstein & Ødegaard 1999). Wittgensteins syn på språket bygger på at det å delta og lære språk, er å delta i *språkspill*. Man lærer ords betydning kun gjennom praktisk bruk – ved å delta i samtaler, eller det han benevner språkspill. For å bruke et eksempel, hadde man ifølge Wittgenstein ikke forstått betydningen av ordet “stol” hadde man ikke hørt ordet brukt i ulike kontekster og slik sett klart å skaffe seg et bilde av hva en stol er. Wittgenstein mener at man ikke forstår språk gjennom definisjoner, men gjennom å lære seg det gjennom praktisk bruk. Hvilken betydning man legger i ulike ord, avhenger av hvilke språkspill man har deltatt i. Men siden språkspill er sosialt konstruert, kan de også modifieres sosialt, og det er dette som må skje gjennom gjensidig læring og kommunikasjon mellom brukere og systemutviklere. Å designe er å konstruere nye språkspill, det vil si at det dannes en syntese av brukerens og systemutviklerens språk og erfaring. En god måte å modifisere og å skjønne hva andre legger i begreper og beskrivelser de gir, er gjennom praksis. Prototyper er en form for praksis. Gjennom prototyper får brukeren et visuelt eksempel, ikke en språkdefinisjon. Brukere på både TelMont og BibTid beskrev prototyper og visuelle hjelpemidler som svært nyttige i å forstå hva utviklerne mente, og forstå hvordan det endelige produktet ville bli seende ut. De mente det var vanskelig å prøve å forstå kravspesifikasjonen med alle dens fagtermer, men straks de så prototyper, forstod hvordan det endelige produktet ville bli, og klarte å komme med innspill. Ehns teori om Wittgensteins språkspill passer med andre ord svært godt inn i våre resultater, og er et sterkt argument for at prototyping bør benyttes så tidlig som mulig i brukermedvirkningen slik at det er lettere for brukerne å komme med innspill.

#### 4.5.10 Grudin

I den skandinaviske tradisjonen er det en sterk tro på at brukermedvirkning bidrar til å skaffe realistiske forventninger om produktet som skapes (Bjerknes & Bratteteig 1995, Bansler 1989). Grudin mener systemutviklere bør ta brukerne med i systemutviklingen fordi det økende utvalget av produkter på programvaremarkedet har ført til økende forventninger fra brukerne (Grudin 1991). Ettersom det blir flere konkurrerende applikasjoner på markedet, minsker brukernes toleranse for dårlige brukergrensesnitt. Dermed blir brukerens mening viktig for å utvikle et salgskraftig brukergrensesnitt, og å finjustere produksjonen etter hva brukeren vil ha. Å bygge kunnskap om brukeren på intuisjon og indirekte metoder blir inadekvat fordi utviklerne ofte er ingeniører med et annet kunnskapsnivå om datamaskiner enn brukeren

(Grudin 1991). I stedet må man ha en direkte brukerinvolvering som kan bidra til å skape de riktige produktene. Disse teoriene stemmer også overens med våre resultater, der brukerne på TelMont og BibTid mente systemet ble bedre tilpasset dem som følge av brukermedvirkningen, og deres fornøydhets taler for at de ikke hadde for høye forventninger til det endelige produktet – antagelig som resultat at de hadde sett deler av det underveis i utviklingen.

#### 4.5.11 Hilbert og Redmiled

Hilbert og Redmiled mener derimot det er et paradoks mellom hvor riktig en bruker håndterer en applikasjon og brukerens subjektive oppfattelse av brukbarheten på applikasjonen (Hilbert & Redmiled 1999). Dette taler *mot* at man bør involvere brukerne i utviklingen, men samsvarer dårlig med våre resultater. Hilbert og Redmiled mener at brukere som scorer høyt på tester om hvor godt de behersker applikasjonen, ofte rapporterer problemer med brukergrensesnittet selv om problemene som rapporteres åpenbart ikke har forhindret brukeren i å utføre vedkommendes oppgaver. Når de blir spurt om en forklaring på dette, sier de noe sånt som: *“Vel, det var lett for meg, men jeg tror at andre kunne ha blitt forvirret.”* Slike evalueringer er med andre ord basert på brukernes subjektive oppfattelser av seg selv, og viser seg ofte å være ubegrunnede. På motsatt hold har man brukere som viser store vansker med å håndtere den samme applikasjonen, men som rapporterer at brukergrensesnittet er godt designet og lett å bruke. Når disse blir bedt om å forklare hvorfor de synes dette, sier de noe sånt som: *“Noen med mer erfaring enn meg ville antagelig tatt dette mye hurtigere.”* Et resultat av dette blir dermed at potensielt viktig feedback fra brukere har vansker med å bli rapportert, mens potensielt misledende respons fra brukere ikke har problemer med å bli rapportert. Man kjenner ennå ikke til slike problemer verken på TelMont eller BibTid, men lignende problemstillinger kan vise seg når produktet har vært i bruk en stund i de to organisasjonene.

#### 4.5.12 Hohmann

Hohmann (Hohmann 1997) argumenterer for at strukturen i en utviklerorganisasjon reflekteres i designet på produktene organisasjonen lager. Han mener at dersom utviklerorganisasjonen har godt organiserte kommunikasjonskanaler og er godt strukturert, vil dette reflekteres i designet på produktene de lager. En velorganisert organisasjonsstruktur skaper en delt mening blant utviklerne om hvordan jobben bør utføres, og dette kan igjen føre til at produktene de skaper blir velstrukturerte. En parallell til dette kan tegnes, nemlig at man burde ha en formell endringshåndtering som for eksempel den som er foreslått i neste kapittel. En god struktur og veldefinerte kommunikasjonsveier for et endringsønske kan bidra til å strukturere endringshåndteringen, og unngå at den kommer ut av kontroll.

### 4.5.13 Forsyth

Forsyth (Forsyth 1999) viser gjennom empirisk materiale at det å planlegge en strategi for hvordan man skal utvikle produkter (for eksempel prosessforbedring) bidrar til bedre verbal kommunikasjon, økt motivasjon og større fleksibilitet når man skal utføre planlagte oppgaver. En parallell kan dras til endringshåndtering, der det å planlegge en prosess for endringshåndtering i fellesskap kan være fordelaktig både for kommunikasjonen, motivasjonen og fleksibiliteten rundt slik at de ikke kommer ut av kontroll. Planleggingen kan for eksempel bidra til at de ulike partene blir mer fleksible i forhold til hverandre, og gir hverandre større spillerom.

## 4.6 Mulige feilkilder

I dette delkapitlet beskrives mulige feilkilder oppgaven kan ha.

En fare ved å utføre prosessforbedringstiltak basert på empiriske studier slik denne oppgaven gjør, er at empiriske studier ikke nødvendigvis er en tilstrekkelig kilde for å forstå nytten av en prosessmodell (Ott, Kinnula, Seaman & Wohlin 1999). En årsak er at empiriske studier gjøres i en kontekst. Det er ikke alltid en forsker får tilgang til informasjon om hele konteksten et case har foregått i, fordi bedriften som har utført prosjektet kan ønske å holde opplysninger konfidensielle av hensyn til firmaets politikk. Det er derfor en risiko at viktige opplysninger som kunne forandret forskerens forståelse av caset holdes tilbake. Det er vanskelig å avgjøre hvorvidt viktige opplysninger er blitt holdt tilbake i de tre prosjektene studert i denne oppgaven.

En fare ved å fokusere hovedsaklig på kvalitative metoder slik denne oppgaven har gjort, er at det er vanskelig å generalisere resultatene (Cornford & Smithson 1996). Kvalitative studier er vanskeligere å summere eller forenkle enn kvantitative studier fordi mange medvirkende faktorer må beskrives og taes hensyn til for å beskrive de ulike resultatene korrekt. En forutsetning for å få riktige data er at personene som ble intervjuet husker begivenheter relativt nøyaktig. Det krever også at tilstrekkelig informasjon er tilgjengelig i datamaterialet for at begivenhetene kan rekonstrueres av forskerne. I tillegg er det ganske naturlig at forskjellige personer som har deltatt i et utviklingsprosjekt har forskjellige "fakta" og meninger om prosjektet, og legger ulik betydning i begreper de bruker. Dette kan skyldes at de har forskjellig erfaringsgrunnlag og har hatt forskjellige roller i prosjektet. Videre kan resultatmaterialet tolkes på forskjellige måter, basert på fortolkerens egne erfaringer, perspektiver, hypoteser og teorier. Resultatene kan dermed ikke gi den vitenskapelige gyldigheten som formelle eksperimenter eller kvantitative undersøkelser gir (Repstad 1998, Kitchenham et al. 1995, Arisholm 2001, Ott et al. 1999). De vil ikke nødvendigvis være gyldig for et stort antall utviklingsteam innenfor et vidt tidsrom. Dette betyr likevel ikke nødvendigvis at



slike kvalitative studier er vitenskapelig uinteressante (Repstad 1998). Resultatene i oppgaven er relatert til eksisterende teori og etablerte begreper, og støtter dermed opp under – samt kaster kritisk lys over – relatert litteratur.

Som i de fleste andre studier, er *ekstern* validitet vanskelig å oppnå. Ideelt sett bør de valgte prosjektene være “typiske” webutviklingsprosjekter for å sikre at erfaringene kan bli generalisert utover de studerte prosjektene. For å redusere inntrykket av slike trusler, er det redegjort detaljert for prosjektene slik at leseren selv har mulighet til å vurdere om erfaringene er relevante for andre prosjekter.

#### 4.6.1 Mulige feilkilder i erfaringsrapporter

Erfaringsrapporter er rapporter som skrives i etterkant av et prosjekt, og oppsummerer hendelser og erfaringer fra prosjektet. Dette resultatkapitlet kan karakteriseres som en erfaringsrapport. Erfaringsrapporter prøver å gi pekepinner på hva andre kan lære fra prosjektet som ble gjennomført. Dette medfører at erfaringsrapporter baseres på historiske data; for eksempel på enkeltindividets hukommelse. Nyttan av en erfaringsdatabase vil avhenge av at man gjennomfører prosjekter relativt likt fra gang til gang slik at man har noe konkret å anvende erfaringene på. Til illustrasjon kan man si det er nyttig å anvende en relativt repeterbar og definert prosess, slik som for eksempel Genovaprosessen søker å være.

Arisholm et. al påpeker at historiske data til en viss grad er nyttige, men at innsamling mens noe skjer er mer effektivt og gir sannere beskrivelser (Arisholm, Anda, Jørgensen & Sjøberg 1999). Historiske data kan fort ha en dårligere kvalitet og gyldighet. Dette er i tråd med hva lærebøker i gruppedynamikk mener; mennesker glemmer eller overrasjonaliserer ofte sine handlinger når de ser på dem i ettertid (Forsyth 1999). Dessuten trenger ikke styrken av en opplevelse – hvor mye tid man tror man har brukt på en handling – nødvendigvis korrelere med virkeligheten. Dersom man gjør noe som man opplever som kjedelig og langtekkelig, kan denne opplevelsen i styrke fortone seg som sterkere, og derfor mer tidkrevende enn andre oppgaver man liker bedre, som man synes tiden “flyr” når man gjør (Forsyth 1999, Jørgensen & Sjøberg 2000). Dersom man faktisk måler den reelle tidsbruken, kan en “kjedelig” oppgave i virkeligheten ha tatt kortere tid enn den “morsomme”, selv om det følte som om den kjedeligste oppgaven tok mest tid. Hvis det for eksempel opplevdes som voldsomt frustrerende å lage flytdiagrammer, tror man ofte denne aktiviteten tok lengre tid enn den i virkeligheten tok, fordi man i utgangspunktet syntes det var en aktivitet det ikke var interessant å bruke mye tid på. Dette kan være en trussel ved å foreta en slik erfaringsinnhenting. En mulig feilkilde kan være at de involverte parter kan ha synes endringshåndteringen har tatt lang tid fordi den var kjedelig, ikke fordi det nødvendigvis tok så lang tid. Det foreligger ingen empirisk belegg for

å hevde at det var endringshåndteringen som gjorde at TelMont-prosjektet ble forsinket, annet enn de subjektive opplevelsene til prosjektdeltakerne, som altså kan være feilaktige dersom de egentlig syntes disse aktivitetene var uinteressante.

Jørgensen og Sjøberg har gjennom empiri funnet grunnlag for at man *ikke nødvendigvis* lærer av erfaring. To empiriske studier av forfatterne viser til problemer med å lære av erfaring som er relevante for denne oppgaven (? , Jørgensen & Sjøberg 2001):

- Samme hendelse oppleves forskjellig av ulike mennesker. Hva man opplever er sterkt influert av hva man forventer å finne (Jørgensen & Sjøberg 2000). Denne faren kan delvis overkommes ved at man inkluderer flere ulike perspektiver i erfaringsinnhenting, noe som er blitt gjort i erfaringsinnhenting i denne oppgaven.
- Det at man er i stand til å formulere en regel ut i fra erfaringene er ofte tilstrekkelig for å slutte at det finnes en regel, selv om det ikke finnes nok empiri som beviser at regelen er gyldig (Jørgensen & Sjøberg 2000). Dette er et generelt problem ved casestudier (Kitchenham et al. 1995). Å tro at man ser mønstre der mønstre ikke finnes er en risiko for denne oppgaven. Det er funnet et mønster om at endringshåndtering lett blir ukontrollert i evolusjonære prosesser, og det er derfor foreslått en måte å løse dette på. Men problemet med en ukontrollert endringshåndtering kan vise seg å være et mønster som er funnet der det i virkeligheten ikke finnes noe mønster.
- Konteksten for et prosjekt er avgjørende for utfallet, og kan gjøre erfaringer verdiløse. Endringer i betingelser kan også gjøre historiske erfaringer ugyldige. Dette er faktorer som er relevante for en erfaringsoverføring fra resultatene i denne oppgaven til andre lignende prosjekter, og som ikke kan kontrolleres.
- Mennesker tenderer mot å bruke deterministiske regler (IF <a> THEN <b>) i stedet for propabilistiske regler (IF <a> THEN 20 % sannsynlighet for <b>). Dette innebærer at man kan tenke “siden jeg ikke nådde målet for prosjektet, estimerte jeg feil”, i stedet for å tenke at man er innenfor sannsynlighetsgrensen for å nå målet. En propabilistisk tenkning vil medføre at man i stedet ser at man ofte kommer noe lavere eller høyere enn sannsynligheten for å nå målet man satte seg. En deterministisk tenkning kan ha gjort seg gjeldende i prosjektledernes dom over endringshåndteringen på TelMont. At endringshåndteringen tok lang tid, finnes det ingen empiri på, kun prosjektledernes frustrerte utsagn. Kanskje var ikke endringshåndteringen så ille som prosjektlederne mente, fordi deres tenkning muligens var deterministisk.

- Lang erfaring gjør ikke nødvendigvis at man er bedre til å forutsi uventede problemer (Jørgensen & Sjøberg 2001). Empiri fra en norsk databedrift viser en relativt svak korrelasjon mellom graden av erfaring og hvor bra man klarte å forutsi uventede problemer som oppstod i utviklingen. Dette kan medføre at en slik erfaringsinnhenting som er beskrevet i dette kapitlet kan være meningsløs fordi det tilsynelatende ikke virker som om man lærer av erfaring.



## Kapittel 5

# En verktøyprototype for endringshåndtering

Dette kapitlet fokuserer på en prosessbeskrivelse og verktøystøtte for hvordan endringer bør håndteres i evolusjonære webutviklingsprosjekter. Dette skyldes at det var problemer rundt endringshåndtering som pekte seg ut som særlig signifikante under erfaringsinnhenting.

Kapitlet ønsker å besvare problemstilling 2:

***Problemstilling 2:** Hvilken prosessbeskrivelse og hvilken verktøystøtte synes hensiktsmessig for endringshåndtering i evolusjonær webutvikling, og hvilke innspill gir dette til Genova-verktøyet?*

Verktøyprototypen for endringshåndtering har som mål at kommunikasjonen mellom de ulike partene i en webutviklingsprosess skal bli enklere, mer strukturert, bedre dokumentert og ta hensyn til alle partene i et utviklingsprosjekt. Med dette menes at kommunikasjonen rundt endringsønsker ikke bare skal foregå mellom utviklere og brukere, men også kunne følges av prosjektledere hos kunde- og utviklerorganisasjon. Fordi partene i et webutviklingsprosjekt gjerne er situert på ulike lokasjoner, ble det ansett som særlig viktig å lage verktøyprototypen med et webgrensesnitt slik at det var lett å aksessere for alle de involverte. Verktøyprototypen er bygget rundt en database som skal ta vare på endringsønskene etter hvert som de kommer inn, og slik både dokumentere endringsønskene og gjøre dem lett tilgjengelige for både kunde- og utviklerorganisasjon. Verktøyprototypen baseres på en prosess som er bygget rundt at også kunde skal godkjenne endringsønskene, og at det skal utarbeides en spesifisering på hvert enkelt endringsønske slik at det vurderes for hver tilfelle om det er forsvarlig med hensyn på estimer og økonomi å implementere dem.

Først i kapitlet er motivasjonen for å lage verktøyprototypen, etterfulgt av en UML-beskrivelse av prosessen verktøyprototypen følger. Deretter følger en mer teknisk beskrivelse av verktøyprototypen. Etter dette følger en be-

skrivelse av hvordan utviklingen av verktøyprototypen foregikk, før det til slutt er en drøfting av relatert litteratur.

## 5.1 Motivasjonen for å lage verktøyprototypen

Retningslinjene fra forrige kapittel ga opphav til utviklingen av en prosessbeskrivelse og en prototype på et verktøy som støtter endringshåndtering i webutviklingsprosjekter. Prosessbeskrivelsen og prototypen ble motivert av det erfaringene viste var et reelt behov for en mer formalisert måte å takle endringshåndtering på gjennom de ulike kommunikasjonspartene (kunde, leverandør og sluttbrukere) i en webutviklingsprosess. Verktøyprototypen vil videreutvikles og innføres som en integrert del av Generas verktøy Genova i nær framtid, og er et svar på underproblemstilling 2, der formålet er å finne ut om erfaringene kan gi innspill til verktøydelen av Genova.

Formålet med verktøyprototypen for endringshåndtering er:

- Å strukturere kommunikasjonen slik at man ikke utelukker en av partene i en utviklingsprosess.
- Å ta hensyn til det økonomiske perspektivet rundt endringshåndtering. Dette gjøres ved at utviklerorganisasjonen lager en spesifisering på endringsønsket. Kundeorganisasjonen vurderer så ut i fra spesifiseringen om dette er noe de vil påkoste.
- Å gjøre verktøyprototypen lett tilgjengelig for sluttbrukere og kundeorganisasjon. Dette gjøres ved å situere verktøyprototypen på web og dermed gjøre det uavhengig av installasjon, lokasjon og tid. Eksterne organisasjoner trenger dermed ikke installere verken databaseserver, webserver eller klientapplikasjoner, og får alltid tilgang på nyeste versjon. Det er også en lav terskel for å lære seg verktøyprototypen fordi det brukes en webbrowser, og skjemaene har et grensesnitt som er kjent fra lignende websider for skjemautfylling.
- Å kunne betjene et ubegrenset antall brukere. Det eneste som utviklerorganisasjonen trenger å distribuere, er et passord for å gi en ny bruker tilgang. Dette medfører at vilkårlig mange sluttbrukere kan delta i utviklingen, noe som kan være positivt i en evolusjonær utvikling der man prøver å finne sluttbrukernes *egentlige* behov.
- Å motivere sluttbrukerne. Det kan være inspirerende for en sluttbruker å se at et endringsønske han har foreslått blir fulgt opp, noe som kan føre til at han/hun yter en ekstra innsats i videre produktutvikling eller i innføringen av verktøyprototypen.
- Å dokumentere endringer som gjøres underveis i prosjektet. En slik dokumentasjon hadde vært nyttig for både TelMont og BibTid. På TelMont førte manglende dokumentasjon av endringsønskene til en stor

diskusjon der utviklerorganisasjonen måtte tilbakebetale noen av timene de hadde fått betalt for. Utviklerne på TelMont mente en bedre dokumentasjon av alle endringene som var foretatt ville skapt større forståelse hos kunde for at estimatene sprakk og produktkostnadene ble høyere enn forventet. På BibTid hadde man dessuten hatt nytte av å dokumentere endringer som var gjort i diskusjoner med sluttbrukerne fordi brukerne stadig gikk tilbake og på nytt ville diskutere endringsønsker de tidligere hadde diskutert. Utviklerne mente en dokumentasjon av endringer som allerede var gjort ville forhindre at diskusjoner rundt de samme endringsønskene ble tatt på nytt.

- Å gjøre det lettere for prosjektlederne hos kunde- og utviklerorganisasjon å ha oversikt over utviklingen. Et verktøy for endringshåndtering som er tilgjengelig for både kunde- og utviklerorganisasjon, dokumenterer endringer utviklerne hadde gjort, er i ferd med å gjøre eller vil gjøre i nærmeste framtid selv uten at utviklerne selv må gjøre mye arbeid med å oppdatere kravspesifikasjonen. For utviklerne blir det da lettere å forsvare tidsbruken fordi prosjektlederne har sett at det er en rekke endringer.
- Å forsikre seg om at man har de økonomiske midler til å gjennomføre endringen. Kunden i TelMont-prosjektet viste stor frustrasjon over at han ikke ble spurt til råds hvorvidt endringsønskene hadde økonomisk støtte før de ble implementert.

### 5.1.1 Innspill til Genova

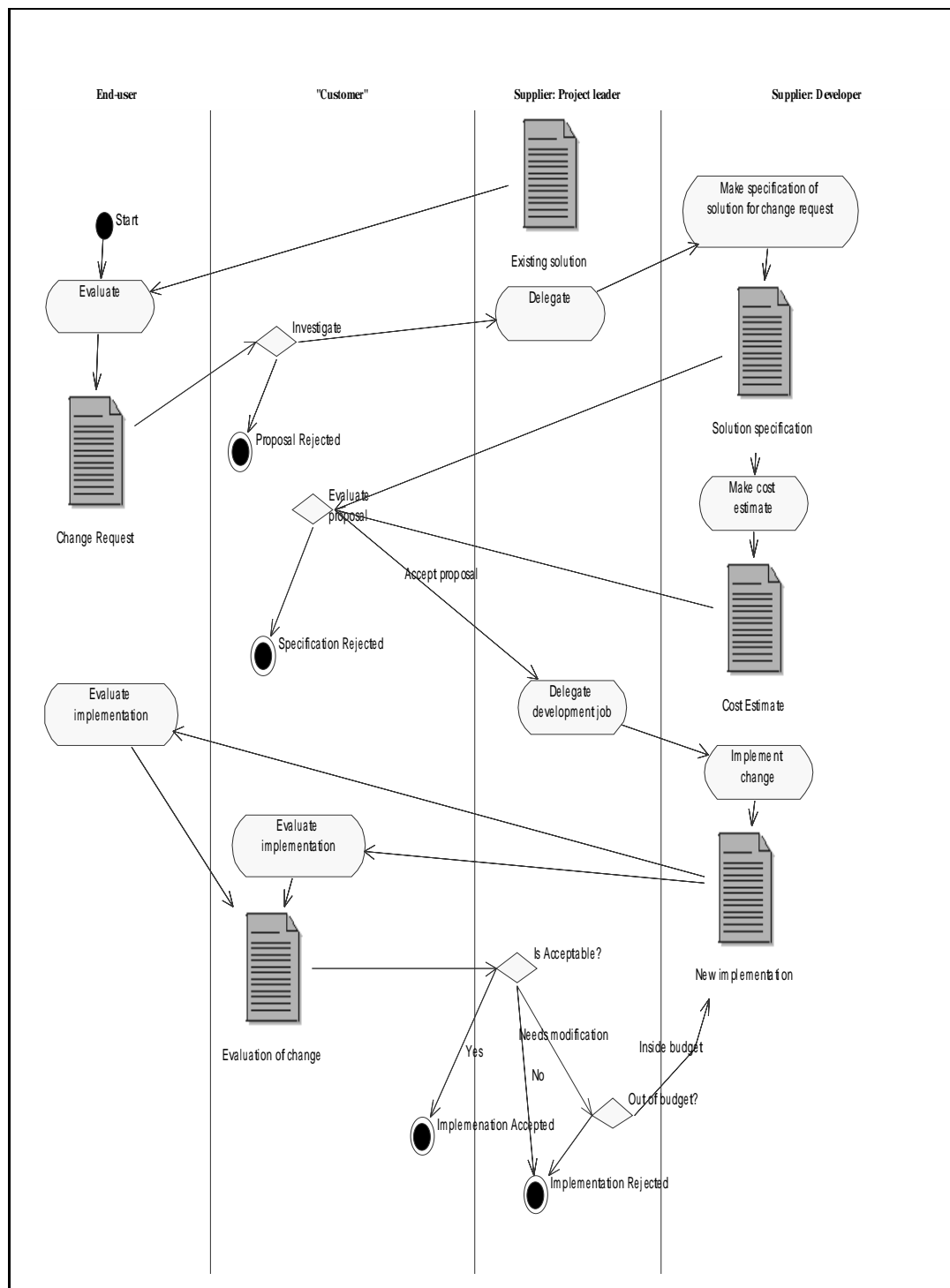
Prosessbeskrivelsen og verktøyprototypen som følger er et innspill til Genova-verktøyet. Hensikten er at prosessbeskrivelsen og prototypen skal evalueres mer detaljert gjennom PROFIT-prosjektet. Basert på denne evalueringen vil prosessbeskrivelsen og verktøyprototypen eventuelt videreutvikles til å bli en del av Genova. En slik evaluering og eventuell integrering er planlagt i Genova i nær framtid.

## 5.2 Prosessbeskrivelse

Aktivitetsdiagrammet og tilstandsdiagrammet presentert er et forslag til en prosessbeskrivelse for endringshåndtering. Diagrammene er hentet fra (Lien et al. 2001). Verktøyprototypen støtter opp under denne prosessbeskrivelsen.

### 5.2.1 Aktivitetsdiagram over arbeidsflyten

Figur 5.1 på forrige side er et aktivitetsdiagram som viser arbeidsflyten i prosessen for endringshåndtering. Et endringsønske oppstår idet en sluttbruker evaluerer den eksisterende løsningen. Produktet av evalueringen er



Figur 5.1: Aktivitetsdiagram for endringshåndtering



et endringsønske. Endringsønsket blir så sendt over til kunden, som ser på det. Kunden kan etter dette enten avvise endringsønsket, noe som fører til at videre utredning av ønsket får en slutt, eller godkjenne det for videre utredning. Dersom ønsket godkjennes for videre utredning, delegerer prosjektleder hos kunde endringsønsket videre til en utvikler i utviklerorganisasjonen. Etter at utvikleren har fått ansvar for et endringsønsket, lager han en spesifisering for det. Denne aktiviteten har som utfall en løsnings-spesifisering, og eventuelt et kostnadsestimat. Denne løsnings-spesifiseringen og eventuelle kostnadsestimat evalueres av kunde, som enten avviser eller godkjenner løsningsforslaget. Dersom kunden avviser løsningsforslaget, avsluttes endringsønsket. Dersom kunde godkjenner løsningsforslaget, kan prosjektleder i utviklerorganisasjonen delegerer implementasjonsansvaret for endringsønsket til en utvikler. Utvikler implementerer så endringer, noe som fører til at det skapes en ny implementert bit av produktet. Denne nye implementasjonen kan så evalueres av sluttbruker og/eller kunde, og resulterer i en endringsevaluering. Prosjektleder hos utviklerorganisasjonen vurderer deretter endringsevalueringen. Dersom endringsimplementasjonen er akseptert og implementasjonen ble akseptert av alle parter, avsluttes arbeidet med endringsønsket. Hvis endringsønsket trenger å modifiseres, blir enten endringsimplementasjonen avvist og endringen faller bort, eller man vurderer om man har råd til å forbedre implementasjonen. Dersom man ikke har råd til å forbedre implementasjonen, avvises endringsønsket, og endringen faller bort. Dersom man har de nødvendige økonomiske midler, sendes endringsimplementasjonen til en utvikler som starter arbeidet med å forfine implementasjonen før den igjen evalueres og er tilbake i den nettopp beskrevne syklusen.

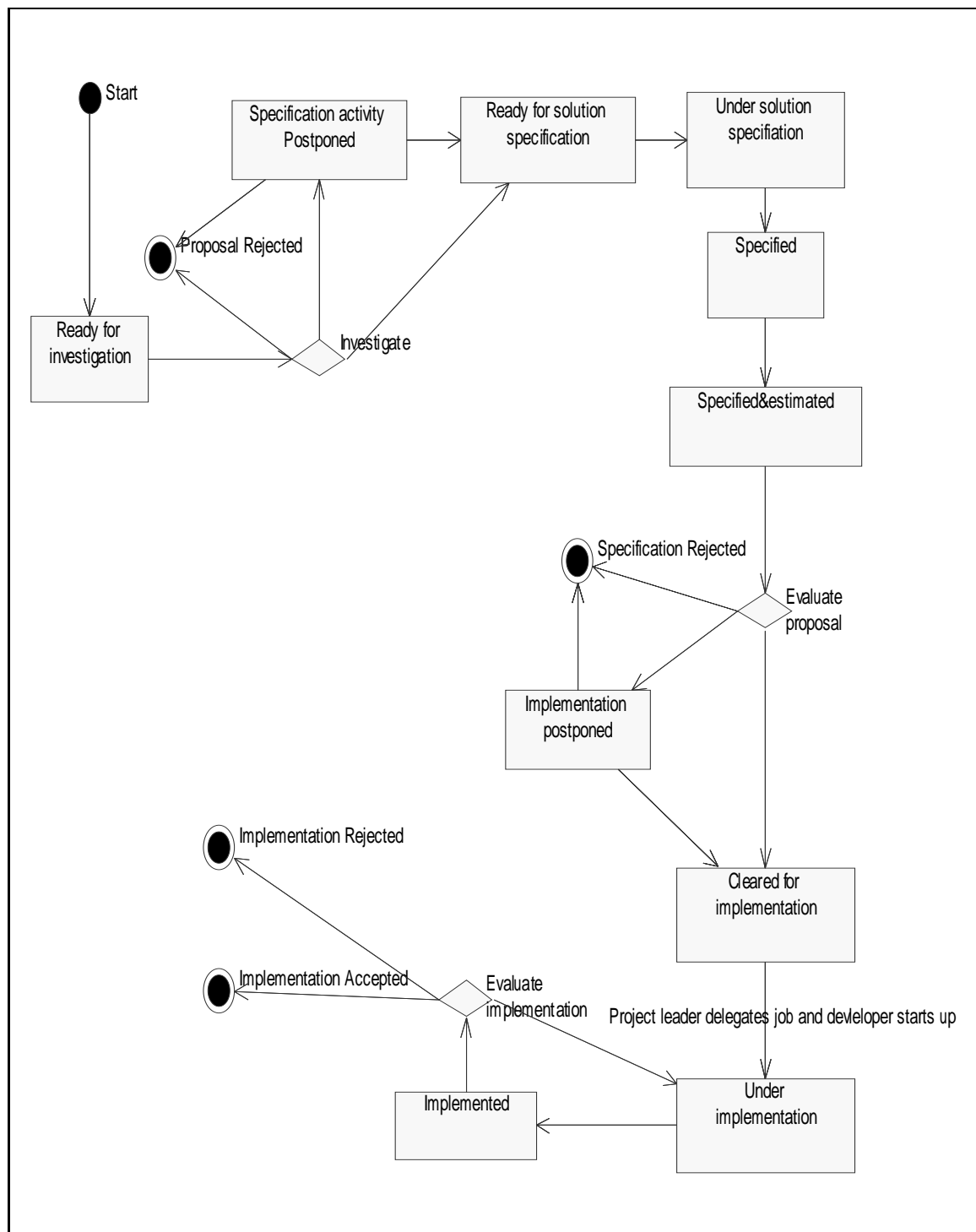
### 5.2.2 Tilstandsdiagram for et endringsønske

Figur 5.2 på neste side er et tilstandsdiagram for et vilkårlig endringsønske. Tilstandsdiagrammet er som følger:

Etter at et endringsønske er oppstått, er det første som skjer at endringsønsket evalueres. Evalueringen av endringsønsket kan føre til en av tre tilstander:

- endringsønsket avvises,
- videre spesifisering av endringsønsket utsettes, eller
- endringsønsket er klart for en løsningsbeskrivelse

Dersom videre spesifisering av et endringsønske utsettes, kan det likevel på et senere stadium komme inn i tilstanden dere det er klart for en løsnings-spesifisering. Det kommer da inn i samme tilstand som endringsønsket kommer inn i dersom det godkjennes umiddelbart etter det har kommet inn,



Figur 5.2: Tilstandsdiagram for et endringsønske

nemlig at det er klart for en løsningsbeskrivelse. Endringsønsket kommer så inn i en tilstand der det er under løsningsspesifikasjon, før det havner i tilstanden der det er spesifisert, og siden inn i tilstanden der det er både spesifisert og estimert. Løsningsforslaget evalueres deretter, slik at endringsønsket nå kan havne i en av tre tilstander:

- spesifikasjonen avvises, og endringsønsket avsluttes
- implementasjonen av endringsønsket utsettes
- endringsønsket klargjøres for implementasjon

Dersom endringsønsket havner i sistnevnte tilstand, delegerer prosjektlederen implementasjonsjobben til en utvikler, og utvikleren starter sin implementasjon. Endringsønsket er nå i en tilstand der det er under implementasjon. Når endringsønsket så kommer i tilstanden “implementert”, evalueres implementasjonen. Endringsønsket havner deretter i en av disse tilstandene:

- Det sendes tilbake til tilstanden der det er under implementasjonen, og fortsetter en ny runde i denne minisyklusen,
- implementasjonen aksepteres og endringsønsket ses som avsluttet, eller
- implementasjonen avvises, og endringsønsket betraktes som avsluttet.

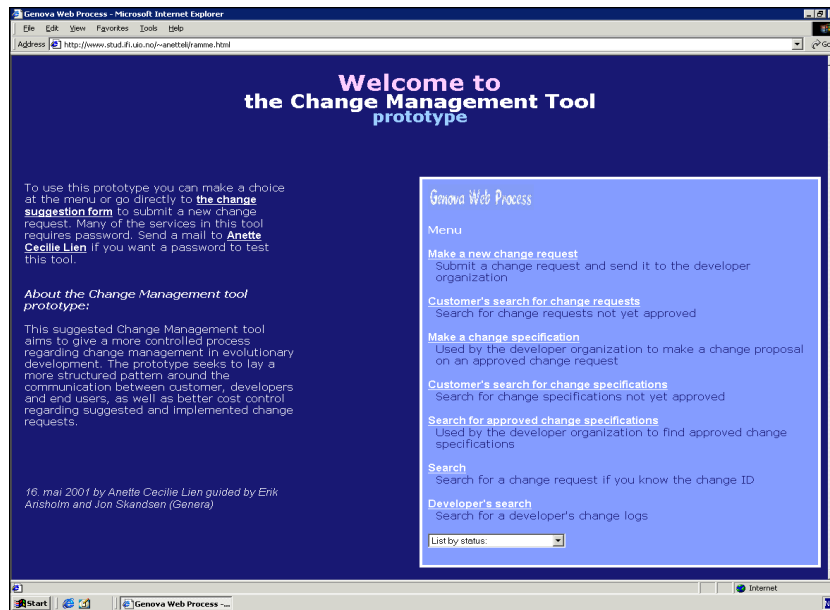
### 5.3 Funksjonalitet og utseende ved verktøyprototypen

En kjørbær leveranse av verktøyprototypen finnes på URL:

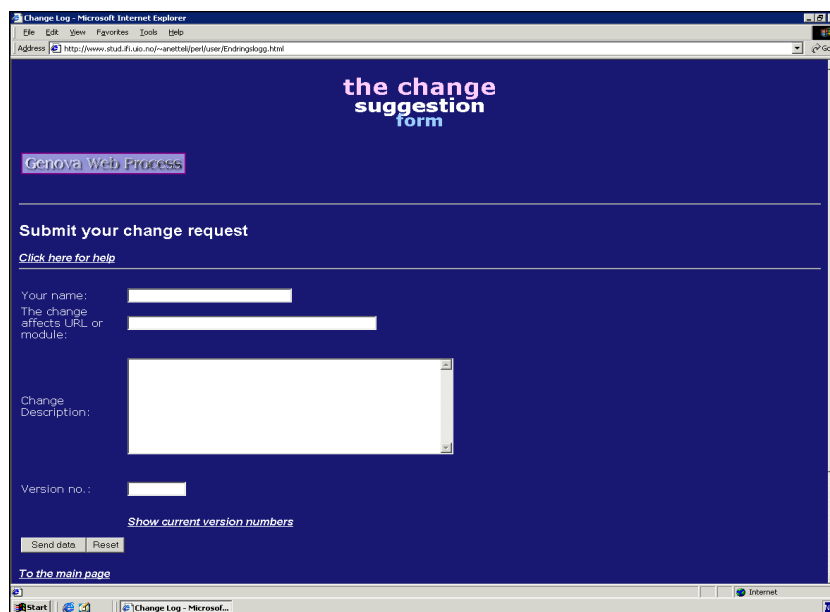
<http://www.stud.ifi.uio.no/~anetteli/ramme.html>

Tjenesten krever brukernavnet (test) og passordet (genova). Med dette passordet får man tilgang som superbruker, det vil si at tilgangen ikke differensieres, slik den gjøres for andre prosjektmedlemmer. Figur 5.3 på neste side er hovedsiden for verktøyet, der menyen til høyre inneholder tilgang til forskjellig funksjonalitet. Figur 5.4 nederst på siden er skjemaet som brukes for å fylle inn et endringsønske, mens figur 5.5 på side 79 er et utsnitt av loggen utviklerne bruker for å loggføre diverse prosjektdata som senere skal brukes i evalueringssystemet. Disse tre skjermbildene er plukket ut blant mange for å gi et inntrykk av ulike sider ved utseendet til verktøyprototypen.

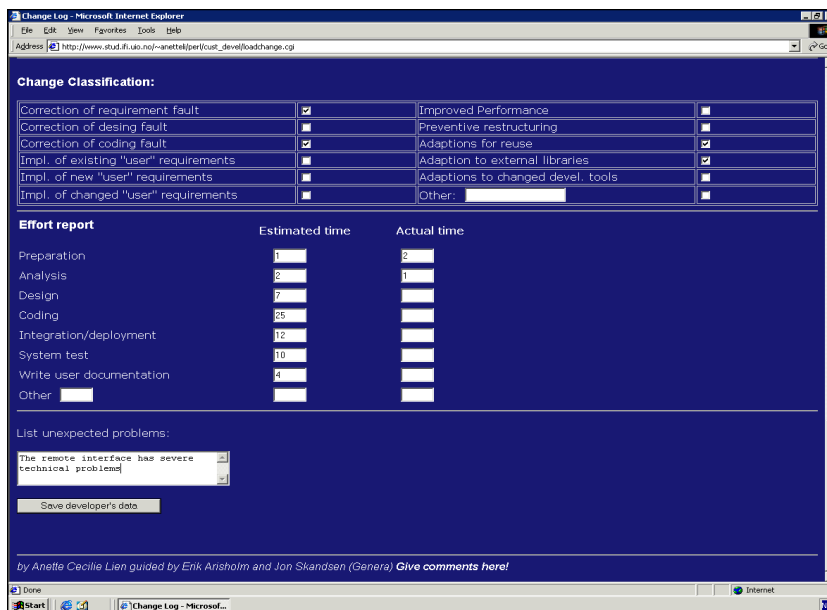
Databasen til verktøyprototypen er bygget rundt en database som tar vare på endringsønskene og tilstandsdataene rundt dem. Databasen er en MySQL-database og kjøres på et Unix-system. Klientsideapplikasjonen er implementert ved hjelp av HTML-sider og Perl/CGI-scripting mot databasen. Alle endringsønsker fra brukeren legges i databasen, og det lages rapporter over endringsønskene. Layouten på siden er bygget opp ved hjelp av CSS-filer.



Figur 5.3: Menyen på verktøyprototypens hovedside



Figur 5.4: Skjema for å fylle inn nytt endringsønske



Figur 5.5: Utsnitt av utviklernes logg

Layouten er en av de tingene ved applikasjonen som har gjennomgått flest forandringer. Først var applikasjonen rammebasert, men da flere testbruker mislikte dette siden de da ikke har anledning til å bokmerke spesielle nettsider inne i applikasjonen, ble dette endret. Også menyen har gjennomgått mange iterasjoner, først og fremst for at den skulle bli mest mulig logisk oppbygd. Slik menyen er i dag, er den bygd opp slik at den følger prosessbeskrivelsen ovenfor i størst mulig grad, det vil si at menyelement 1 kommer før menyelement 2 i prosessen og så videre. Meningen er at endringsønsket følges gjennom å klikke seg nedover i linkrekkefølgen. De ulike brukerne av applikasjonen har differensiert tilgang. Disse partene har tilgang til hver sine deler av applikasjonen:

- Kunde har tilgang til å se på endringsønskene, godkjenne dem for utredning, og godkjenne utredningen etter den var foretatt.
- Prosjektleder hos utviklerorganisasjonen har tilgang til å tilordne ansvaret for et endringsønske (det vil si ansvaret for spesifikasjonen og en eventuell implementasjon) til en utvikler.
- Utviklere har tilgang til å loggføre data om endringsønsket etter hvert som de utviklet på det, for eksempel estimert tid brukt og faktisk tid brukt
- Sluttbrukerne har tilgang til å legge inn et nytt endringsønske

- Alle partene har tilgang til å legge inn endringsønske og til å se oversiktsrapportene for de forskjellige endringsønskene, dvs om hvor langt ønsket var kommet i prosessen (om det var blitt godkjent, om det var spesifisert etc..)

### 5.3.1 Overordnet beskrivelse av verktøyprototypen

Verktøyprototypen følger prosessbeskrivelsene som er beskrevet under forklaringene av modellene beskrevet i Rational Rose. Foruten dette, inneholder verktøyprototypen funksjonalitet for at utviklerne logger data om implementasjonen. Disse loggede dataene er tenkt brukt av forskere i evalueringsoyemed. Dataene som logges er endringsklassifikasjon, estimatrapporter og uventede problemer.

Verktøyprototypen kan generere følgende statusrapporter:

- Rapport over nye endringsønsker som ikke er godkjent
- Rapport over godkjente endringsønsker
- Rapport over utsatte endringsønsker
- Rapport over avviste endringsønsker

Testbruker 4 likte ikke denne menyen. Hun mente det var vanskelig å skjønne om rapportene gjaldt helt nye endringsønsker, eller endringsønsker som allerede var vurdert, og hvor spesifikasjonen skulle godkjennes eller avvises. Dette er en nyttig innvending som vil bli tatt hensyn til i videre arbeid med prototypen.

### Databaseskjemaet

Databasen har databaseskjemaet vist i tabellen under, der primærnøkler er i kursiv.

**Kritikk av databaseskjemaet:** Siden prosessen med å utvikle verktøyprototypen var en evolusjonær prosess, lider strukturen på databaseskjemaene under dette. Nye felter er blitt lagt til og gamle felter er blitt fjernet ettersom endringsønskene fra brukerne ble tatt hensyn til. Muligens kan dette ses som en kritikk av evolusjonær utvikling i seg selv. Evolusjonær utvikling er av blant andre Boehm blitt kritisert for å produsere ustrukturert kode (Boehm 1988). Denne kritikken er blitt eksemplifisert gjennom utviklingen av denne verktøyprototypen.

**CHANGECLASSIFICATION**

<i>ChangeID</i>		Correction_of_req_fault	Correction_of_design_fault
Correction_of_coding_fault		Impl_of_existing_user_requirements	Impl_of_new_user_requirements
Impl_of_changed_user_requirements		Improved_performance	Preventive_restructuring
Adaptions_for_reuse		Adaption_to_external_libraries	Other
Other_value			

**CHANGEDESCRIPTION**

<i>ChangeID</i>	PreviousChangeID	OpenedTime	ClosedTime	ResponsibleDeveloper
-----------------	------------------	------------	------------	----------------------

**CHANGEPROPOSAL**

<i>ChangeID</i>	Proposal	Status	Comments	Project
-----------------	----------	--------	----------	---------

**CHANGESTATUS**

<i>ChangeID</i>	Investigated	Status_bef_invest	Status_of_proposal	Delegated_to_level
-----------------	--------------	-------------------	--------------------	--------------------

**CHANGESUGGESTION**

<i>ChangeID</i>	Suggestor	ChangeDescription	Status	url_module	comments	project	version_no	RequestTime
-----------------	-----------	-------------------	--------	------------	----------	---------	------------	-------------

**EFFORTREPORT**

<i>ChangeID</i>	preparation	analysis	design	coding	integration	system_test	user_doc	other	other_effort
-----------------	-------------	----------	--------	--------	-------------	-------------	----------	-------	--------------

**ESTIMATION**

<i>ChangeID</i>	preparation	analysis	design	coding	integration	system_test	user_doc	other	total
-----------------	-------------	----------	--------	--------	-------------	-------------	----------	-------	-------

**SUBJECTIVEESTIMATION**

<i>ChangeID</i>	exper_task_size	exper_task	resulting_changeability	problems
-----------------	-----------------	------------	-------------------------	----------

Tabell 5.1: Databaseskjemaet som ligger til grunn for verktøyprototypen

## 5.4 Utvikling og evaluering av verktøyprototypen

Arbeidet med å utvikle prototypen foregikk med evalueringshjelp fra fem testbrukere. Testbruker 1 var mest deltagende i den initielle idéfasen av utviklingen av prototypen, samt da verktøyprototypen var ferdigstilt. Testbruker 2 var mest deltagende under implementeringen av prototypen. Testbruker 3 var aktiv under hele prosessen. Testbruker 4 og 5 var aktive etter at prototypen var mer eller mindre ferdigstilt.

Idefasen, det vil si den fasen da prosessbeskrivelsen bak verktøyprototypen ble lagd, foregikk i samarbeid med testbruker 1. En rekke aspekter ved prosessen som skulle ligge til grunn for verktøyprototypen ble diskutert. Særlig ble det diskutert i hvor stor grad man skulle ta hensyn til det økonomiske aspektet ved prosessen, og det ble vurdert dit hen at dette aspektet var viktig. Dette skyldes at det særlig var rundt det økonomiske aspektet at stridighetene rundt endringshåndteringen hadde oppstått på TelMont. Det ble derfor vurdert som sentralt at en spesifisering på endringsønsket (både teknisk og økonomisk spesifisering) skulle gjøres, og at denne spesifiseringen igjen skulle bli godkjent av kunde før endringen ble implementert. Formålet med å inkludere dette i prosessen var å unngå økonomiske kranbler tilsvarende den man fant i TelMont i kommende utviklingsprosjekter.

Den første reaksjonen til testbruker 2 da han ble forklart ideen bak verktøyprototypen, var at et slikt verktøy var akkurat det han trengte. Han hadde lett etter et slikt verktøy uavhengig av vår utvikling av en slik prototype. Han hadde ikke funnet noe verktøy som var passende for hans formål, og hadde i stedet begynt å lage et enkelt system selv i Word. Da han fikk se de første skjermbildene og funksjonaliteten bak, kom han med en rekke konstruktive innspill på ting han synes burde være annerledes. Han tok utgangspunkt i hva han selv ville trenge dersom han skulle ha brukt verktøyprototypen på prosjektet han da var ansvarlig for.

Førsteintrykket til testbruker 2 var at verktøyprototypen virket for omfattende. Han var skeptisk til de mange detaljene, og til overheaden et slikt verktøy ville medføre. Testbruker 2 ønsket seg også differensiert tilgang til verktøyprototypen, fordi han ikke ville at alle parter i utviklingsprosessen skulle ha tilgang til å gjøre de ulike operasjonene. For eksempel ønsket han at alle partene skulle ha tilgang til å legge inn et nytt endringsønske, men at bare kunden skulle få lov til å godkjenne om det skulle implementeres eller ikke. Han ville også at prosjektlederen hos utviklerorganisasjonen skulle ha tilgang til mer funksjonalitet enn utviklerne, for eksempel funksjonalitet for å tilordne et endringsønske til en bruker. Det ble derfor lagd differensiert tilgang.

Testbruker 3 kom på flere møter underveis i utviklingen med en rekke forslag til forbedringer av verktøyprototypen. Han testet særlig brukervennligheten,



og kom med forslag til hvordan brukere mer intuitivt kunne forstå applikasjonen dersom visse endringer ble gjort. Han kom aktivt med endringsønsker basert på skjermbildene som ble vist ham. Alle møtene med testbruker 3 skjedde ved personlig oppmøte, der applikasjonen ble vist til han mens utvikleren var tilstede. På denne måten ble blant annet navigasjonsstrukturen i applikasjonen forandret fordi testbrukeren stanset og stusset på hvordan han skulle komme seg videre. Dette ble det tatt hensyn til i videre implementasjon. Testbruker 3 var særlig aktiv i navigasjonsstrukturen på applikasjonen. Testbrukeren fant også fram til feil ved funksjonaliteten ved å gå utradisjonelle veier gjennom applikasjonen.

Da applikasjonen var ferdigstilt ble den testet av to nye testbrukere med to måneders mellomrom. Den første testbrukeren evaluerte verktøyprototypen til å være nyttig for hennes formål (å *dokumentere* endringer som ble gjort i en applikasjon). Hun fant også noen funksjonalitetsfeil ved å gå veier i applikasjonen som verken utvikleren eller de andre testbrukerne hadde tenkt på. Utvikleren var tilstede da hun testet verktøyprototypen, og noterte seg mentalt feil eller kommentarer. Testbrukeren var svært positiv til verktøyprototypen, og synes ideen bak verktøyprototypen var svært god. Hun la også planer i samråd med prosjektet hun var deltagende i om å bruke verktøyprototypen. Dermed ble det opprettet en egen konto slik at prosjektet kunne bruke verktøyprototypen. Kontoen er imidlertid ikke blitt brukt fordi prosjektet endret kurs og ikke lenger skulle utføre endringer.

Etter to måneder ble det opprettet kontakt med en fjerde testbruker, som i begynnelsen da hun fikk høre om ideen stilte seg interessert i å bruke verktøyprototypen i bedriften Braathens. Men etter en nøyere gjennomgang av prototypen, synes hun den var for kompleks for bedriftens formål. Hun syntes prosessen man skulle følge i verktøyprototypen ble for kompleks og tungrodd, og foreslo en løsning som gikk ut på at man kuttet ut det økonomiske aspektet ved prosessbeskrivelsen. Hun kunne tenkt seg å benytte seg av verktøyprototypen dersom det hadde vært atskillig enklere, med kun ett eller to skjermbilder, og hvor man ikke trengte godkjenning av spesifikasjoner og være i kontakt med kunde for å begynne å implementere et endringsønske. Hun vurderte det derfor som uaktuelt å ta i bruk verktøyprototypen i organisasjonen.

## 5.5 Sammenligning med andre verktøy

Under vil ulike verktøy for endringshåndtering beskrives, og sammenlignet med verktøyprototypen som ble utviklet. Beskrivelsene er basert på demonstrasjonsversjoner av verktøyene som finnes på Internett.

### 5.5.1 Rational ClearQuest

Rationals prosess RUP er støttet av flere ulike verktøy (Kruchten 2000). Verktøyene støtter opp om den evolusjonære prosessmodellen, og ønsker blant annet å støtte endringshåndteringen som følger med hver iterasjon. Unified Change Management (UCM) er Rational Software sin tilnærming til å håndtere endringer i systemutvikling. UCM spenner fra kravspesifikasjon til endelig leveranse, og prøver å definere hvordan man skal håndtere endringer i krav, designmodeller, dokumentasjon, komponenter, testing og kildekode. UCM ønsker å sammenveve aktivitetene brukt for å planlegge og å holde oversikt over prosjektframgangen og hvilke elementer som er under endring. UCM-modellen realiseres av både en prosess og verktøy. Rationals produkter ClearCase og ClearQuest er teknologiene som støtter opp om UCM. ClearCase er et versjonshåndteringssystem som holder orden på dokumenter (deriblant kildekode) som produseres underveis i prosjektutviklingen. Verktøyet ClearQuest er ment for å assistere endringshåndteringen i et prosjekt. ClearQuest tar vare på prosjektets oppgaver, feil og endringsønsker underveis i utviklingen, samtidig som det holder rede på utviklingen i prosjektet.

Rational ClearQuest er et kommersielt verktøy for feil- og endringshåndtering som hvis ønskelig kan gis et webgrensesnitt (Rational 2001). Det kan også brukes som et eget verktøy for intern bruk i prosjekter uten at man bruker det via web. ClearQuest fungerer i et klient-server-miljø. Verktøyet har ferdige skjemaer man kan bruke som mal for endringsønsker, samtidig som verktøyet tilbyr funksjonalitet for å tilpasse skjemaene etter bedriftens egne ønsker. ClearQuest er integrert med Rationals andre produkter, deriblant Rational Unified Process, samt en rekke Microsoft-produkter (for eksempel Microsoft Access, Word, Microsoft SQL server). ClearQuest har også et eget administrasjonsverktøy for å tilpasse verktøyet til organisasjonens bruk. I administrasjonsverktøyet kan man sette opp brukerkontoer, endre skjemaene som brukes for å sende inn endringsønsker, spesifisere tilgangskontrollen, velge parametere for å generere rapporter etc.

Når man sender inn et endringsønske i ClearQuest, tillatter man brukeren å fylle inn prioritet og alvorlighetsgrad. Man kan også konfigurere ClearQuest slik at det automatisk sendes e-post til prosjektmedlemmer når det kommer inn et nytt endringsønske eller et endringsønske forandres.

ClearQuest kan også generere grafer over endringsønskene. Man kan for eksempel få generert et søylediagram over arbeidsmengden til de forskjellige

utviklerne, eller et søylediagram som rangerer endringsønsker etter tilstand og alvorlighetsgrad. Det er også mulig å generere rapporter over endringsønskene. Rapporter kan baseres på parametere som man velger fra gjennom predefinerte nedtrekksmenyer. Administrator kan også lage nye rapportformater i ClearQuests eget redigeringsverktøy.

Endringsønskene gjennomgår flere forskjellige tilstander i ClearQuest-verktøyet. Den første tilstanden et endringsønske kommer inn i etter det har blitt sendt inn, er at det blir tilordnet en utvikler. Mens utvikleren jobber med endringsønsket, er det i tilstanden “åpnet”. Når endringsønsket er gjort ferdig av utvikleren, setter han/hun det i tilstanden “løst”. Man kan her for eksempel velge at hver gang et endringsønske er løst, sendes en e-post til kvalitetssikringsteamet. Etter at et kvalitetssikringsteam har gått gjennom og sett på endringsønske, og eventuelt godkjent, settes det som “avsluttet”. Man kan selv regulere tilgangen det enkelte prosjektmedlem skal ha til de forskjellige operasjoner, slik at en utvikler for eksempel ikke kan gå inn og sette et endringsønske til “avsluttet”. Det er relativt enkelt å søke etter tidligere feil eller endringsønsker i databasen, og man kan enten bruke forhåndsdefinerte spørringer mot databasen, eller komponere sine egne.

### 5.5.2 Elementool

Elementool er i likhet med ClearQuest et kommersielt verktøy for feil- og endringshåndtering. Hovedfokus er lagt på å holde orden på feil og forslag til forbedringer som oppstår underveis i et prosjekt, og etter at produktet er ferdigstilt. Elementool er kun lokalisert på web, og det må betales for tilgang til verktøyet. Flere IT-bedrifter abonnerer på en slik tilgang, noe som bekrefter behovet for en strukturert endringshåndtering. Skjemaet for å sende inn en ny feil eller et endringsønske karakteriseres ved at den som sender inn feilmeldingen, også må spesifisere OS, CPU og browser selv, samt har tilgang til å tilordne prioritet, alvorlighetsgrad og utvikler. Det kan også lages rapporter over feil og endringsønsker. Rapportene genereres som en liste over feil, og diverse informasjon om feilen. Også i Elementool har en feil/endringsønske flere tilstander. Når en feil/endringsønske er rettet opp av en utvikler, markeres den som “fixed”. Etter at feilen deretter er rettet opp, testet og godkjent, markeres endringen som “closed”. I Elementool finnes det også et eget redigeringsverktøy slik at man kan tilpasse rapportgenerering og skjemaer etter spesifikke behov i bedriften, samt gi de ulike brukerne differensiert tilgang til applikasjonen.

### 5.5.3 Telelogic ChangeSynergy 4.0

TeleLogic ChangeSynergy er et kommersielt, webbasert verktøy for endringshåndtering (Telelogic 2001). Endringene er ment å komme fra både interne

og eksterne kilder. På samme måte som i verktøyprototypen, er et av målene bak ChangeSynergy at utviklingsteam, avhengig av lokasjon og størrelse, i større grad skal komme i kontakt med kundens behov. Ønsket er at man skal kunne samarbeid på tvers av geografisk plassering. ChangeSynergy ønsker også å støtte endringshåndtering i komponentbasert utvikling, slik at endringer kan gå på tvers av produkter, komponenter og utviklingsteam.

I ChangeSynergy finnes det ferdige maler for skjemaer og rapportgenerering. Rapportene er integrert med Microsoft Excel og Word, slik at de kan lastes ned og lagres i et av disse Microsoft-formatene. Det finnes også et redigeringsverktøy der man kan legge til roller, spesifisere arbeidsflyten bak endringshåndteringen, lage egne spørringer mot databasen og definere rapporter.

ChangeSynergy er rollebasert, med to definerte roller (“bruker” og “administrator”). Man kan også legge til egne roller dersom ønskelig. Man gis forskjellig tilgangskontroll ut i fra hvilken rolle man har. De forskjellige rollene kan gjennom redigeringsverktøyet gis tilgang til å tilordne et endringsønske til en utvikler, eller til å overvåke et endringsønske. En endring gjennomgår forskjellige tilstander, og man kan generere rapporter basert på status. ChangeSynergy har ferdige maler for å generere grafer og trendanalyser over endringsønskene, i likhet med ClearQuest. I ChangeSynergy kan man sende filvedlegg med et endringsønske. Man kan også lage et endringsønske som er relatert til tidligere endringsønsker, i motsetning til i verktøyprototypen.

ChangeSynergy tilrettelegger for replikering mot andre databaser for endringshåndtering. Endringsønsker kan legges inn i en database og replikeres over i en annen. Dette innebærer at et endringsønske kan legges inn i én database, men utføres av et annet utviklingsteam med en annen database. Det er også mulig å dele et endringsønske inn i flere deler, og distribuere ulike deler av endringsønsket til ulike databaser.

#### **5.5.4 Request Tracker**

Request Tracker er gratisvare fra GNU GPL, og er et system for å håndtere spørsmål for en gruppe av brukere (Vincent & Alderdice 2001). Systemet er veldig generelt, og det kan brukes til endringshåndtering, som supportsystem for kundebehandlere, feilhåndtering eller for alle mulige formål der en gruppe mennesker ønsker å holde oversikt over en liste av oppgaver. Request Tracker har et webgrensesnitt, slik at hver sluttbruker lett kan se statusen på sin forespørsel via en webside. Request Tracker er ment for små til mellomstore selskaper. Endringshåndteringen i Request Tracker er basert på at en bruker sender endringsønsket som e-post. Som kvittering får brukeren et elektronisk merke som er kjennetegnet på endringsønsket. Dette merket brukes i all senere korrespondanse rundt ønsket, samtidig som det sendes

e-post til valgte prosjektmedlemmer for å varsle om det nye endringsønsket. Et prosjektmedlem tar ansvar for ønsket og sender et svar. Svaret blir lagret i databasen, samtidig som det blir sendt til brukeren på e-post. Prosjektmedlemmet løser så endringsønsket.

I Request Tracker kalles et endringsønske for en “billett”. Billetter kan ha ulik status: ny, åpnet, utsatt, løst og dødt. Med utsatt menes at det ikke finnes noen løsning på problemet enda, og det er utsatt inntil videre for en senere løsning. Men dødt menes at endringsønsket ble avvist av prosjektteamet. De ulike brukerne av systemet har ulik tilgang til systemet. I Request Tracker deles brukerne inn i privilegerte og ikke-privilegerte brukere. Det er kun de privilegerte brukerne som har tilgang til systemet via web. Når en vilkårlig bruker sender inn en e-post med et endringsønske, blir den en uprivilegert bruker. Den har da ikke tilgang til webgrensesnittet, men mottar e-post om hva som skjer videre med endringsønsket. Brukerne kan ikke tilordne en prioritet til et endringsønske, noe som taler for at dette verktøyet er ment for en kommunikasjon med sluttbrukere i større grad enn de andre verktøyene. Et endringsønske kan tilordnes en prioritet av de privilegerte brukerne. Det er fem nivåer av prioritet: lav, middels, høy, kritisk, fatal. I Request Tracker har brukeren som sender inn endringsønsket dermed ikke så mye ansvar, slik at Request Tracker tydelig er myntet på sluttbrukere av produktet som er eksterne til utviklerorganisasjonen. På dette punktet er Request Tracker svært likt verktøyprototypen.

Også i Request Tracker genereres rapporter. Det er også mulighet til å søke etter endringsønsker basert på mange ulike kriterier. I Request Tracker finnes det også funksjonalitet for å knytte enkelte endringsønsker til hverandre og til eksterne databaser. På grunn av den åpne kildekoden man finner i en slik gratisvare, er det enkelt å selv forbedre verktøyet og tilpasse det egne behov.

### 5.5.5 Sparxsystems

Sparxsystems Enterprise Architect er et utviklingsverktøy basert på UML. Sparxsystems Enterprise Architect har også et innebygd endringshåndteringsverktøy (Sparxsystems 2001). Dette verktøyet konsentrerer seg om endringer som kommer fra brukeren *etter* at produktet er levert for installasjon, det vil si i vedlikeholdsfasen til produktet. Siden denne forskjellen er en såpass markant forskjell fra verktøyprototypen, beskrives ikke Sparxsystems nærmere.

### 5.5.6 RMT, CCC

RMT og CCC er to gratisvarer for endringshåndtering som ligner mye på GNUs Request Tracker. Verktøyene er webbasert, og tar vare på og viser endringsønsker på svært lik måte til Request Tracker.

### 5.5.7 Funksjonalitet ved de andre verktøyene som mangler i verktøyprototypen

- Integrasjon med andre verktøyet (for eksempel Microsoft- eller Rationals produkter, andre databaser for endringshåndtering).
- Konfigurasjon slik at det automatisk sendes e-post til prosjektmedlemmer når det kommer inn et nytt endringsønske eller statusen på et endringsønske forandres.
- Et redigeringsverktøy for å legge til brukere, for å sette opp parametere for rapportgenerering, for å styre tilgangskontroll, for å velge felter på skjemaene som skal fylles inn, og for å legge til egne roller.
- Tilfredsstillende søk etter endringsønsker i databasen. Slik verktøyprototypen er i dag, kan man kun søke etter endrings-ID, og antagelig vil man også ønske å søke etter nøkkelord.
- Funksjonalitet for å generere grafer og statistisk materiale over endringsønskene.
- Mulighet for å komponere egne spørringer mot databasen, ikke bare predefinerte spørringer.
- Funksjonalitet for å sende filvedlegg med et endringsønske.
- Mulighet for å dele et endringsønske inn i flere deler.
- Tilordning av prioritet eller alvorlighetsgrad på et endringsønske. Mulighet for slik informasjon gjøres kun gjennom et skriftlig kommentarfelt kunden kan fylle ut.
- Funksjonalitet for å knytte enkelte endringsønsker til hverandre og til eksterne databaser.
- Tilrettelegging for replikering mot andre databaser for endringshåndtering.

### 5.5.8 Hva verktøyprototypen har som de andre mangler

- Å ta hensyn til kommunikasjonen mellom sluttbrukere, kunde og utviklere, er noe som er unikt for verktøyprototypen. Grunnlaget for å lage verktøyprototypen var da også at denne kommunikasjonen sviktet i TelMont og tildels også i BibTid, samt at et verktøy ble sett på som en løsning for å strukturere denne kommunikasjonen. Et mål med verktøyprototypen er at ingen av partene utelukkes. Ingen av de andre verktøyene har tatt hensyn til at verktøyet skal støtte kommunikasjonen mellom *alle* de tre partene.
- Verktøyprototypen tar hensyn til det økonomiske perspektivet rundt endringshåndtering gjennom at utviklerorganisasjonen lager en spesifisering på endringsønsket som kunden vurderer. Ingen av de andre

verktøyene tar hensyn til det økonomiske perspektivet, som nettopp var det mest problematiske i TelMont.

- Verktøyprototypen involverer kunden i prosessen rundt hvorvidt endringsønskene skal implementeres eller ikke. Verktøyprototypen sikrer dermed at utvikleorganisasjonen forsikrer seg om at de har de økonomiske midler fra kunden til å gjennomføre endringen. Ingen av de andre verktøyene forsikrer seg om dette, noe som kan medføre at endringer kan utføres som det egentlig ikke er økonomisk dekning for.
- Verktøyprototypen inkluderer sluttbrukerne som aktive deltakere. I flere av verktøyene som er undersøkt, kunne brukeren selv sette prioritet, alvorlighetsgrad og delegere ansvaret for sitt endringsønske. At en bruker selv kan sette slik informasjon, overlater mye ansvar til den som sender inn endringsønsket. Dette kan vitne om at endringshåndteringsverktøyene er best egnet for internt bruk hos utviklerorganisasjonen i prosjektet. Det vil være uforsvarlig å la en sluttbruker delegere sitt endringsønske til en utvikler, slik enkelte av verktøyene gjør. Skjemaene som skal fylles inn krever også mye teknisk spesifisering, noe som gjør dem vanskelig tilgjengelig for sluttbrukere uten mye teknisk datakunnskap. I verktøyprototypen er informasjon som skal fylles inn på skjemaene for et endringsønske skåret ned til det basale for at det skal være enklest mulig for en sluttbruker å fylle dem ut.
- Et endringsønske kan følges kontinuerlig av sluttbrukerne, uten at de må vente på e-post eller en muntlig statusrapport. Det kan være inspirerende for en sluttbruker å se hvordan endringsønsket han/hun har foreslått blir fulgt opp. Ingen av de andre verktøyene legger til rette for sluttbrukernes deltagelse på samme måte.
- Et verktøy for endringshåndtering som er tilgjengelig for både kunde- og utviklerorganisasjon gjør at også kunden kan følge endringene underveis, og på denne måten ha oversikt over utviklingen selv om utviklernes egen dokumentasjon og oppdatering av kravspesifikasjonen er manglende. Ingen av de andre verktøyene legger til rette for at kunden skal kunne ha oversikt over utviklingen på samme måte.
- Felles for verktøyene som her er evaluert og verktøyprototypen, er at de alle inneholder ulike tilstander som endringsønskene gjennomgår. Tilstandene er nokså like, og betegner hvorvidt endringsønsket er nytt, om det er under implementering, om det er ferdig implementert og om det er godkjent/testet. Men verktøyprototypen skiller seg fra de evaluerte verktøyene ved at det også inneholder tilstander for hvorvidt endringsønsket er godkjent for spesifisering av kunde, hvorvidt det er spesifisert, og hvorvidt spesifiseringen er godkjent av kunde. Tilstandene oppstår fordi kunden involveres i beslutningsprosessene, i motsetning til de andre verktøyene som ble evaluert.

## 5.6 Avsluttende bemerkninger

Prosessbeskrivelsen og verktøyprototypen som er beskrevet i dette kapitlet, er et innspill til Genova-verktøyet. Ønsket er at den formaliserte prosessen og verktøystøtten kan hjelpe med å strukturere endringshåndtering, og hjelpe til bedre kontroll over det økonomiske aspektet ved endringshåndtering.

Ifølge flere av testbrukerne kan det virke som om verktøyprototypen er for kompleks og omfattende, med for mange skjermbilder og for mange hensyn å ta til de ulike kommunikasjonspartene i endringshåndteringen. Selv om det på bakgrunn av erfaringene fra TelMont virker som om det økonomiske aspektet burde taes hensyn til, — og med det at de ulike kommunikasjonspartene involveres — virker det som om testbrukerne syntes dette blir for mye ekstraarbeid.

Verktøyprototypen mangler mye viktig funksjonalitet som de andre verktøyene har. Dette må sees i lys av en mangel på ressurser, da et verktøyforslag skulle gjøres innenfor rammen av en enkelt hovedoppgave. Derimot har verktøyprototypen også flere aspekter som de andre verktøyene ikke har tatt høyde for. Håpet er at ideene bak verktøyet skal gjøre det til et nyttig produkt som tilfører endringshåndteringen elementer de andre verktøyene ikke inkorporerer. På denne måten kan det bli et alternativt verktøy i de tilfeller der man antar at kommunikasjonen rundt endringshåndtering i en evolusjonær prosess kan komme ut av kontroll.



## Kapittel 6

# Konklusjon

Denne oppgaven har vært en del av PROFIT-prosjektet, og har bidratt gjennom å gi innspill til Genova webprosess. Bidraget bestod av:

- Kartlegging av erfaringer fra tre evolusjonære webutviklingsprosjekter. Så vidt forfatteren av denne oppgaven bekjent, er det ikke gjort en tilsvarende erfaringskartlegging rundt evolusjonære webutviklingsprosjekter tidligere.
- Utarbeidelse av retningslinjer for prosessforbedring basert på erfaringskartleggingen.
- Utvikling av en prosessbeskrivelse for endringshåndtering basert på retningslinjene.
- Utvikling av en kjørbare verktøyprototype for endringshåndtering basert på prosessbeskrivelsen. Det spesielle med denne verktøyprototypen er at den støtter kommunikasjonen mellom sluttbrukere, kunde og utviklere på en måte som ingen andre verktøy for endringshåndtering på det kommersielle markedet gjør. Dessuten tar verktøyprototypen hensyn til det økonomiske perspektivet rundt endringshåndtering; kunden involveres i de økonomiske beslutningene rundt hvorvidt et endringsønske skal implementeres eller ikke, noe man ikke finner i tilsvarende verktøy på markedet.

### 6.1 Erfaringsinnhenting og retningslinjene for prosessforbedring

Erfaringsinnhenting var basert på 15 intervjuer med brukere, kunder og utviklere i tre evolusjonære webutviklingsprosjekter, og dannet grunnlaget for utarbeidelsen av retningslinjer for prosessforbedring. Disse retningslinjene er et innspill til Genova webprosess. Erfaringsinnhenting viste at webutvikling lider under bruken av umoden webteknologi og at det kreves

kompetanseoppbygging i utviklerorganisasjonene for å utvikle for web. Særlig påvirket disse to faktorene estimatene. De påvirket også hvordan man bør gjøre initiering og prototyping i webutviklingsprosjekter. Prosjektene opplevde også at webutvikling har manglende funksjonalitet i forhold til det utvikling av tradisjonelle applikasjoner tilbyr, og at dette kan by på problemer i forhold til hvor brukervennlig grensesnittet kan gjøres.

Erfaringsinnhentingene viste videre at endringshåndtering i evolusjonær utvikling kan være en utfordring. Grunnen til at endringshåndtering er en spesielt stor utfordring ved evolusjonær utvikling kan være at man i denne type utvikling ønsker å ta hensyn til sluttbrukernes innspill *underveis* i utviklingen for å tilpasse produktet til deres behov. Dette står i kontrast til en utvikling der man kun er i kontakt med sluttbrukerne mens man skriver kravspesifikasjonen eller etter at man har levert fra seg det ferdige produktet.

Resultatene viste også at brukerne syntes det var vanskelig å bidra med kommentarer og endringsforslag før de hadde sett konkrete resultater av prosjektet, dvs. før de hadde sett prototyper og deler av den kommende applikasjonen. Straks de hadde blitt vist deler av applikasjonen, kom de derimot med en rekke endringsforslag til utviklerorganisasjonen. Endringsforslagene ble positivt mottatt og i størst mulig grad implementert, noe som gjorde at brukerne selv syntes de fikk det produktet de hadde ønsket seg i første omgang. Et problem var derimot at brukerne etter hvert kom med så mange endringsønsker at endringshåndteringen kom ut av kontroll fordi utviklerorganisasjonene manglet formelle rutiner for å håndtere endringsønskene. Å gjenvinne kontroll ble etter hvert den største utfordringen under utviklingen. Særlig gjaldt dette prosjektet TelMont.

På TelMont medførte mangelen på kontroll over endringshåndteringen både estimatoverskridelser og frustrasjon blant prosjektledere i utviklerorganisasjon og hos kunde. Kunden som på sin side skulle betale utviklingen, følte seg tilsidesatt fordi kommunikasjonen rundt endringshåndteringen stort sett foregikk mellom brukere og utviklere, uten at kunden ble konsultert.

Resultatene indikerte dermed at endringshåndtering kan være vanskelig uten en formell prosess, fordi deler av nødvendig kommunikasjon kan bli forbigått. Som et forslag til løsning på disse problemene, ble det utarbeidet en prosessbeskrivelse og en prototype på et endringshåndteringsverktøy som støtter opp under prosessbeskrivelsen. Prosessbeskrivelsen og verktøyforslaget er ment å formalisere endringshåndteringen, og sikre kommunikasjonen mellom alle de tre parter i webutviklingsprosessen (kunde, utviklere og brukere) slik at ingen av partene blir tilsidesatt.

## 6.2 Prosessbeskrivelsen og verktøyprototypen for endringshåndtering

Utviklingen av prosessbeskrivelsen og verktøyprototypen var i seg selv en evolusjonær prosess som ble gjort i samarbeid med fem testbrukere. Prosessbeskrivelsen er basert på at kunde skal godkjenne endringsønskene, og at det skal utarbeides en spesifikasjon på hvert enkelt endringsønske slik at det vurderes for hvert tilfelle om det er forsvarlig med hensyn på estimer og økonomi å implementere dem. Verktøyet støtter opp under prosessbeskrivelsen. Verktøyprototypen er bygget rundt en database som skal ta vare på endringsønskene etter hvert som de kommer inn, og slik både dokumentere endringsønskene og gjøre dem lett tilgjengelige både for kunde- og utviklerorganisasjon. Fordi partene i et webutviklingsprosjekt gjerne er situert på ulike lokasjoner, ble det ansett som viktig å lage verktøyet med et webgrensesnitt slik at det var lett å aksessere for alle de involverte.

Det finnes mange verktøy for endringshåndtering på markedet, og mange av dem har mye viktig funksjonalitet som verktøyprototypen mangler. Manglende funksjonalitet ved prototypen er blant annet et redigeringsverktøy for å legge til brukere av applikasjonen, samt grafisk graf-generering over ulike typer endringsønsker. Prosessbeskrivelsen og verktøyprototypen innehar derimot kvaliteter som andre verktøy på markedet mangler. At prosessen formaliserer og støtter kommunikasjonen mellom brukere, kunde og utviklere er spesielt for verktøyprototypen. Dette var et område som særlig sviktet i prosjektene som ble studert. Prosessbeskrivelsen og verktøyprototypen tar videre hensyn til det økonomiske perspektivet rundt endringshåndtering i større grad enn de andre verktøyene for endringshåndtering på markedet. Også en økonomisk styring hadde det vært et sterkt behov for under de studerte prosjektene. Prototypen involverer videre kunden i større grad rundt avgjørelser av hvorvidt endringsønsker bør implementeres eller ikke, noe kunden på TelMont etterlyste under erfaringsinnhenting.

Flere av testbrukerne syntes derimot verktøyprototypen var for kompleks og omfattende — at den tok for mye hensyn til de ulike aspektene ved endringshåndtering. Det kan virke som om prosessbeskrivelsen legger opp til en for omfattende prosedyre rundt endringshåndtering som kanskje ikke vil kunne fungere i et utviklingsmiljø som er preget av stramme tidsskjemaer og raske avgjørelser.

## 6.3 Videre arbeid

De generelle prosessretningslinjene som er foreslått på basis av erfaringene vil videre bli evaluert på kommende webutviklingsprosjekter som en del av det nasjonale forskningsprosjektet PROFIT. Evalueringen er neste steg i en

iterativ prosessforbedrings-syklus. Videre arbeid kan bestå av å utvikle flere retningslinjer for prosessforbedring i evolusjonære webutviklingsprosjekter, og samle disse generelle erfaringene i prosessmodellen Genova webprosess. Dette arbeidet er gjenstand for nye hovedoppgaver innen PROFIT.

Verktøyprototypen ble godt mottatt av Genera, og vil antagelig bli re-implementert og forbedret som en del av deres CASE-verktøy Genova. Før en reimplementering og forbedring av prototypen gjøres, vil det være hensiktsmessig med en grundigere evaluering av prototypen. En måte å gjøre dette på, kan være å innføre prototypen i et pilotprosjekt og gjennom observasjon og intervjuer med brukerne av prototypen utrede positive og negative aspekter ved både innføringen og prototypen. Endringsønsker til prototypen kan sendes inn gjennom bruk av prototypen. Også dette arbeidet kan føre til nye hovedoppgaver innen PROFIT.

# Bibliografi

- Arisholm, E. (2001), Empirical Assessment of Changeability in Object-Oriented Software, PhD thesis, Departments of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo.
- Arisholm, E., Anda, B., Jørgensen, M. & Sjøberg, D. (1999), Guidelines on conducting software process improvement studies in industry, *in* '22nd IRIS Conference (Information Systems Research Seminar In Scandinavia)'
- Arisholm, E., Benestad, H. C., Skandsen, J. & Fredhall, H. (1998), Incorporating rapid user interface prototyping in object-oriented analysis and design with genova, *in* 'NWPER'98 Nordic Workshop on Programming Environment Research', pp. 155–161.
- Arisholm, E., Skandsen, J., Sagli, K. & Sjøberg, D. (1999), Improving an evolutionary development process — a case study, *in* 'Proceedings of the European Software Process Improvement Conference'99', pp. 9.40–9.50.
- Axmark, D. & Widenius, M. (2000), *The MySQL Reference Manual*, MySQL AB.  
**URL:** <http://www.mysql.com/doc/>
- Bansler, J. (1989), 'System development research in scandinavia', *Scandinavian Journal of Information Systems* **1**, 3–20.
- Bødker, K., Kensing, F. & Simonsen, J. (2000), *Professionel IT-forundersøgelse — grundlaget for bæredyktige IT-anvendelser*, Samfundslitteratur.
- Bjerknes, G. & Bratteteig, T. (1995), 'User participation and democracy. a discussion of scandinavian research on system development', *Scandinavian Journal of Information Systems* **7**(1), 73–98.
- Boehm, B. (1988), 'A spiral model of software development and enhancement', *IEEE Computer*, *21*(5) pp. 61–72.

- Braa, K. & Vidgen, R. (1999), 'Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research', *Accounting Management and Information Technologies* **9**, 25–47.
- Bratteteig, T. (1983), *Kommunikasjon i systemutvikling*, Master's thesis, Institutt for Informatikk, University of Oslo.
- Bratteteig, T. (1997), Mutual learning. enabling cooperation on systems design, in 'Proceedings of IRIS'20', pp. 1–20.
- Bratteteig, T. & Stolterman, E. (1997), 'Design in groups — and all that jazz', *Computers and Design in Context* pp. 289–316.
- Castro, E. (1998), *Perl and CGI for the World Wide Web: Visual Quickstart Guide*, Addison-Wesley.
- Cloyd, M. H. (2000), Creating a user-driven development process: In web time, in 'Doing Business On the Web — 6th Conference on Human Factors and the Web'.
- Cornford, T. & Smithson, S. (1996), *Project Research in Information Systems — a students guide*, Macmillan Press Ltd.
- Cotton, T. (1996), 'Evolutionary fusion: A customer-oriented incremental life cycle for fusion', *Hewlett-Packard Journal* **47**(4), 25–38.
- Dalland, O. (1997), *Metode og oppgaveskrivning for studenter*, Universitetsforlaget, Oslo, Norge.
- Dumas, J. S. & Redish, J. (1999), *A Practical Guide to Usability Testing*, Intellect Books.
- Ehn, P. (1993), Scandinavian design: On participation and skill, in D. Schuler & A. Namioka, eds, 'Perspectives on Systems Design: Participatory Design', Lawrence Erlbaum Associates Inc, New Jersey, pp. 41–77.
- Elementool (2001), *Elementool Demo*, Elementool Inc.  
**URL:** <http://www.elementool.com/cgi-bin/demo/1.asp>
- Elmasri, R. & Navathe, S. (2000), *Fundamentals of Database Systems*, 3rd edn, Addison-Wesley.
- Emam, K., Quintin, S. & Madhavji, N. (1996), 'User participation in the requirements engineering process: An empirical study', *Requirements Engineering* **1996**(1), 4–26.
- Faulkner, C. (1998), *The Essence of Human-Computer Interaction*, Ashford Colour Press Ltd.

- Floyd, C. (1984), A systematic look at prototyping, *in* 'Approaches to prototyping(editors)', Springer-Verlag, pp. 1–19.
- Forsyth, D. (1999), *Group Dynamics*, third edn, Wadsworth publishing company.
- Gilb, T. (1988), *Principles of Software Engineering Management*, Addison-Wesley,.
- Grudin, J. (1991), 'Obstacles to user involvement in software product development, with implications for cscw', *International Journal of Man Machine Studies* **34**(3), 435–452.
- Halvorsen, K. (1993), *Å forske på samfunnet – en innføring i samfunnsvitenskapelige metoder*, third edn, Bedriftøkonomens forlag.
- Hilbert, D. & Redmiled, D. (1999), 'Collecting user feedback and usage data on a large scale to inform software development', *Technical Report UCI-ICS-99-41* .
- Hohmann, L. (1997), *Journey of the Software Professional – A sociology of Software Development*, Prentice Hall.
- Homer, A., Ullmann, C. & Wright, S. (1997), *Instant HTML – programmer's reference*, Wrox Press Ltd.
- Humphrey (1989), *Managing the Software Process*, Addison-Wesley, Essex, England.
- Johnsen, T. E. (2000), – ring meg! en studie av ungdom og mobiltelefoni, Master's thesis, Instutt for Kulturstudier, University of Oslo.
- Jørgensen, A. H. (1984), On the psychology of prototyping, *in* 'Approaches to prototyping(editors)', Springer-Verlag, pp. 105–122.
- Jørgensen, M. & Sjøberg, D. I. K. (2000), The importance of not learning from experience, *in* 'Proceedings of the European Software Process Improvement Conference'00', pp. 2.2 – 2.8.
- Jørgensen, M. & Sjøberg, D. I. K. (2001), 'Learning from experience in a software maintenance environment', *Journal of Software Maintenance (accepted for publication)* .
- Järvinen, P. (1999), *On Research Methods*, Opinpaja Oy.
- Kitchenham, B. (1996), *Software metrics: Measurement for Software Process Improvement*, Blackwel.
- Kitchenham, B., Pickard, L. & Pfleeger, S. L. (1995), 'Case studies for method and tool evaluation', *IEEE Software* **12**(4), 52–62.

- Knight, K., ed. (1989), *Participation in Systems Development*, GP Publishing, Columbia, Maryland.
- Kruchten, P. (2000), *The Rational Unified Process — An Introduction*, 2nd edn, Addison-Wesley.
- Lehman, M. & Belady, L. (1985), *Program Evolution, Processes of Software Change*, Academic Press, London, U.K.
- Lichter, H., Schneider-Hufschmidt, M. & Zullighover, H. (1994), ‘Prototyping in industrial software projects — bridging the gap between theory and practice’, *IEEE Transactions on Software Engineering* **20**(11), 825–832.
- Lien, A., Arisholm, E. & Skandsen, J. (2001), Evolutionary development of web-applications — lessons learned, in ‘Proceedings of the European Software Process Improvement Conference’01’, pp. 4.14–4.32.
- Lientz, B., Swanson, E. & Tompkins, A. (1978), ‘Characteristics of application software maintenance’, *Communications of the ACM* **21**(6), 466–471.
- Lindgaard, G. (1994), *Usability Testing And System Evaluation. A guide for designing usefull computer systems*, Champman and Hall.
- Linger, R. (1993), ‘Cleanroom software engineering for zero-defect software’, *15th International Conference on Software Engineering (ISCE’93)* pp. 2–13.
- Mathiassen, L. (1995), ‘Prototyping and specifying: Principles and practices of a mixed approach’, *Scandinavian Journal of Information Systems* **7**(1), 55–72.
- Mathiassen, O. & Forsth, L. (1988), *Prosjekt*, Origo Forlag AS.
- May, E. L. & Zimmer, B. A. (1996), ‘The evolutionary development model for software’, *Hewlett-Packard Journal* **4**(47), 39–45.
- Myers, M. (1997), ‘Qualitative research in information systems’, *MIS Quarterly* **2**(21), 241–242.  
**URL:** <http://www.misq.org/misqd961/isworld/.MISQDiscovery>
- Negrino, T. & Smtih, D. (1999), *Javascript for the World Wide Web (Visual Quickstart Guide)*, Peachpit Press.
- Næss, L. (2001), *User Guide*, Genera AS.
- Ott, L. M., Kinnula, A., Seaman, C. & Wohlin, C. (1999), ‘The role of empirical studies in process improvement’, *Empirical Software Engineering* **4**, 381–386.



- Pfleeger, S. L. (1998), *Software Engineering: Theory and Practice*, Prentice Hall.
- Preece, J. (1993), *A Guide to Usability: human factors in computing*, Addison-Wesley.
- Rational (2001), *ClearQuest Demo*, Rational Software Corporation.  
**URL:** <http://www.rational.com/tryit/clearquest/index.jsp>
- Repstad, P. (1998), *Mellom nærhet og distanse*, Universitetsforlaget, Oslo, Norge.
- Royce, W. (1970), 'Managing the development of large software systems: Concepts and techniques'.
- Saarinen, T. & Saaksjarvi, M. (1990), 'The missing concept of user participation', *Scandinavian Journal of Information Systems* **2**, 25–42.
- Schuler, D. & Namioka, A., eds (1993), *Participatory design: Principles and Practices*, Lawrence Erlbaum Associates Inc, New Jersey.
- Schwartz, R. & Christiansen, T. (1997), *Learning Perl*, O'Reilly.
- Seaman, C. (1999), 'Qualitative methods in empirical studies of software engineering', *IEEE Transactions on Software Engineering* **25**(4).
- Sefland, J. P. (1998), Brukermedvirkning i utvikling av en gruppevareapplikasjon – teori møter praksis, Master's thesis, Instutt for Informatikk, University of Oslo.
- Sjøberg, D. (1993), Thesaurus-Based Methodologies and Tools for Maintaining Persistent Application Systems, PhD thesis, University of Glasgow.
- Sjøberg, D. & Arisholm, E. (2000), 'Notat: Klyngen for inkrementell evolusjonær utvikling'.  
**URL:** <http://www.geomatikk.com/spiq/temagrupper/klyngenotat/klyngenotat%20040100%20IEU.html>
- Sommerville, I. (1995), *Software Engineering*, fifth edn, Addison-Wesley, Essex, England.
- Sparxsystems (2001), *Sparxsystems Enterprise Architect*, Sparx Systems Pty. Ltd.  
**URL:** [http://www.sparxsystems.com.au/ea\\_downloads.htm](http://www.sparxsystems.com.au/ea_downloads.htm)
- Telelogic (2001), *Telelogic ChangeSynergy*, Telelogic AB.  
**URL:** <http://www.telelogic.com/products/synergy/changesynergy.cfm>
- Thomassen, L. (1996), Bedriftsintroduksjon ved hjelp av data assistert opplæring, Master's thesis, Norges Handelshøyskole, Bergen.

- Ulsund, T. (2000), 'Spiq'.  
**URL:** *URL på* <http://www.geomatikk.no/spiq/>
- Ulsund, T. (2001), 'Profit'.  
**URL:** <http://193.215.229.22/>
- Vincent, J. & Alderdice, M. (2001), *Request Tracker*, GNU GPL.  
**URL:** <http://www.fsck.com/projects/rt/download.html>
- Vogt, Y. (2001), 'Nordmenn i verdensteten', *ComputerWorld number 1* .  
**URL:** <http://www.computerworld.no/cwi.nsf/f/F8189616.01.2001>
- Wall, L., Christiansen, T. & Schwartz, R. L. (1996), *Programming in Perl*, 2nd edn, O'Reilly.
- Walsham, G. (1993), *Interpreting Information Systems in Organizations*, John Wiley and Sons Inc., West Sussex, England.
- Ward, S. & Kroll, P. (2000), 'Building web solutions with the rational unified process: Unifying the creative design process and the software engineering process'.
- Wiig, I. (1994), Metodisk støtte for brukermedvirkning og kreativitet i systemutvikling, Master's thesis, Instutt for Informatikk, University of Oslo.
- Wittgenstein, L. & Ødegaard, T. (1999), *Tractatus logico-philosophicus*, Gyldendal, Oslo, Norway.
- Yin, R. (1995), *Case Study Research – design and methods*, Vol. 5, second edn, Sage Publications.
- Zamperoni, A., Gerritsen, B. & Bril, B. (1995), 'Evolutionary software development: An experience report on techical and strategic requirements.', *Technical report TR-95-25* .

## Tillegg A

# Evolutionary Development of Web-applications — Lessons Learned

**Anette Cecilie Lien and Erik Arisholm**

*Industrial Systems Development*

*Department of Informatics*

*University of Oslo*

{anetteli;erika}@ifi.uio.no

**Jon Skandsen**

*Genera AS*

jsk@genera.no

### *Om vedlegget*

*Dette vedlegget er en artikkel som omhandler resultatene presentert i kapittel fire i denne hovedfagsoppgaven. Artikkelen er publisert i "Proceedings of the European Software Process Improvement Conference'01".*

### **Abstract**

This paper describes experiences from evolutionary web development projects. These experiences suggest several, simple process guidelines. These guidelines will subsequently be evaluated on new development projects, as the next step in a process improvement cycle. We studied three web development projects out of which two included user participation. Users, customers and developers were interviewed. The users had difficulties in making contributions before they were shown some parts of the product. Once prototypes were developed, the users continually had new suggestions and change requests during the development process. However, the change management came out of control because lacking formal routines for handling the change requests. Furthermore, the results show that web-technology is immature. This may in turn affect the way project initiation, estimation and prototyping should be performed.

## A.1 Introduction

Genera AS is a vendor of Genova, which is an advanced CASE tool for object-oriented analysis and design, dialog modelling, and automatic application generation and database generation (Arisholm et. al 1998). In conjunction with the development of the Genova tool, we are also developing the evolutionary Genova process. Such evolutionary development processes have been proposed as an efficient way to deal with risks such as new technology (e.g., web-technology) and imprecise or changing requirements (Boehm 1988). The main idea is to resolve risks early by incrementally evolving the system towards completion instead of relying on the traditional “big-bang” waterfall approach (Royce 1970). Thus, an important objective of evolutionary development is to identify the “real” needs of the customer as the system evolves. However, exactly how evolutionary development should be performed to provide the benefits hoped for remains an open research question (Arisholm 2001). It is plausible that the process models will need a significant amount of tailoring depending on various characteristics of the development project at hand.

During the past few years, Genera AS have been involved in many web application development projects. Through the project experiences, it has become apparent that web application development may have many characteristics that require specific process support. This paper describes initial efforts to develop the Genova Web Process, tailored to support evolutionary (i.e., iterative and incremental) development of web applications. More specifically, experiences from three web development projects in which Genera was involved are reported. Based on these experiences, several process guidelines are proposed. We believe these guidelines may be useful for other companies in similar development projects. A simple prototype of a change management tool is also described. This development of this tool is motivated by the observed need to formalise communication paths among stakeholders in evolutionary development projects.

The remainder of this paper is organised as follows. Section 2 gives an overview of process improvement activities in Genera related to the development and empirical evaluation of the Genova process, and gives an overview of the study design. Section 3 describes experiences from the three web development projects. Section 4 recommends a set of process guidelines, based on the experiences from the analysed projects. Section 5 relates the results to existing studies. Section 6 concludes and describes future work.

## A.2 Improving the Genova Process

Initially, the Genova process was defined as a scaled down version of the Rational Unified Process (RUP). We believe that such a “light-weight” evolutionary process will increase the likelihood of success in smaller development projects (Arisholm et al. 1999).

At present, further development of the Genova process is underway. The study described in this paper is a part of a process improvement project funded by a Norwegian industry research project, PROFIT. The goal of a sub-project of PROFIT (in conjunction with Genera) is to develop guidelines for evolutionary development projects in general, but with emphasis on web applications development in

<b>Project Name</b>	<b>TelMont</b>	<b>BibTid</b>	<b>UniBase</b>
<b>Project Manager</b>	Yes	Yes	Yes
<b># Web Developers</b>	2	2	1
<b>Customer</b>	Yes	Yes	No
<b># Users</b>	4	2	None

Tabell A.1: Overview of the interviewees

particular. The guidelines are based on the experiences collected from three web development projects through interviews with (1) developers and project management representing the main contractor, and (2) end-users and management representing the customer. General guidelines will subsequently be incorporated in the Genova Process. Guidelines specific to web development projects will be incorporated in the Genova Web Process. The guidelines will subsequently be evaluated, both qualitatively and quantitatively, on new web development projects. Thus, the experiences presented in this paper describes the results of the first phase of an effort seeking to improve the way web applications should be developed using an evolutionary process.

The following subsections describe the design of the experience collection phase of the Genova improvement project. The study consisted of semi-structured interviews with subjects involved in three web development projects. The following experience collection process was used:

- Project selection
- Subject selection
- Data collection
- Analysis

### A.2.1 Project Selection

The development projects were selected based on relatively recent projects in which Genera was involved, either as a main contractor or as a service provider for the main contractor. Since the experience collection is based on the subjects' recollection of events, selecting fairly recent projects probably improves the quality of the data. Furthermore, the goal was to select quite diverse projects, in an attempt to uncover as many characteristics of web development as possible, including both positive and negative experiences. However, all of the selected projects are web-based systems that are accessible only within the customer organization's intranet. Table A.2 gives an overview of the selected web projects. The TelMont project contributed substantially more to the process guidelines outlined in Section 4 than the other two projects. Note that the project and company names have been changed for confidentiality reasons.

### A.2.2 Subject Selection

Several interviews were conducted with subjects selected to cover different roles in the development projects. In total, 15 persons were interviewed, see table A.1. The

subjects had central roles in the development projects, and consisted of

- developers and project management from the contractor,
- the customer project management, and
- the end-users.

Before the interview sessions, several specific questions were formulated by each of the two researchers. The questions were combined and prioritised to form an interview guide. The interviews were recorded on a tape recorder in order to avoid loss of information. As a starting point for the data collection, the interview guide worked very well. Other, more open-ended questions were also asked. Thus, the interviews were designed not only to elicit the information foreseen, but also unexpected types of information. For the end-users in the TelMont project, a questionnaire consisting of 11 questions was distributed by email. Four out of the six end-users involved in the development project answered the questionnaire.

### **A.2.3 Analysis**

For the largest and most complex project, the TelMont project, each question and answer from the recorded interviews were written down in detail. Although this transcription process is very time consuming, it is in our opinion essential to improve the accuracy and comprehensiveness of the analyses. It is particularly important for the unforeseen types of information, which may be scattered and intermixed within several answers to specific and open-ended questions. Each researcher used the transcribed material to perform the analysis in parallel, and wrote an analysis report based on the data from the interviews. Finally, these reports were combined into a unified analysis report. The experiences from this analysis process clearly show the benefits of performing the analyses in parallel. Each of the two researchers analysed the data using different perspectives resulting in distinct themes or categories of experiences. If only one researcher had analysed the data, important information would have been lost.

For the LibTime and UniBase projects, a less comprehensive analysis method was used. The interviews were still recorded on tape, but no formal transcription or parallel analysis took place, because these projects were smaller and therefore simpler to analyse. All of the analysis reports were given to the interviewees for quality assurance. The experiences described in this paper are based on the revised analysis reports.

### **A.2.4 Threats to Validity**

There are a number of threats to the validity of qualitative studies such as this. The study assumes that the interviewees remember events reasonable well. Even then, different subjects may remember “facts” in different ways, amongst others because they have different experience backgrounds and played different roles in a project. Furthermore, a sufficient amount of information needs to be made available to reconstruct events and subsequently deduce lessons learned. Consequently, a certain amount of interpretation of the data is performed by the researchers. Such interpretations are biased by the researchers’ own experiences and theories. Thus, there are clearly many sources of threats to the internal validity of this study. As

with most case studies, external validity is also difficult to achieve. Ideally, the selected projects should be representative of “typical” web development projects to ensure that the lessons learned can be generalized beyond the studied projects. To reduce the impact of such threats, detailed descriptions of the projects are provided to enable the reader to assess whether the experiences might be of relevance for another project.

## A.3 Experiences from the Web Development Projects

This section provides a detailed description of the studied development projects, outlined in table A.2.

### A.3.1 TelMont

TelMont is a support system used by a telecommunications company. The support system simplifies and automates the work processes required for performing compression and optimisation activities for ISDN telephony hardware. The users of the system are professional engineers. An external company (i.e., the “contractor”) developed the product on a per-hour-basis. The development phase of the project required 8600 person-hours over a nine-month period. The project consisted of a total of 10 to 15 persons, in addition to some external consultants. Specialists in the customer organisation who knew the ISDN compression and optimisation work processes contributed to the analysis and design.

#### *Project Activities and Milestones:*

##### *May 1999 - August 1999 (Inception)*

A pilot project was initiated by the customer in May 1999. The goal of the pilot project was to determine the need for an automated software solution. The deliverable from the pilot project was an analysis report.

##### *August 1999 - October 1999 (Elaboration)*

The requirement specification activities started in August. In this phase of the project, the main work processes that the TelMont system was intended to support were discussed. The work resulted in a requirement specification and an analysis model consisting of a workflow model and some use cases, as well as a simple prototype of the user interface.

##### *October 1999 - March 2000 (Iterative Elaboration and Construction)*

In this phase of the project, a large evolutionary prototyping activity was initiated. The prototype had two purposes: 1) to evaluate the feasibility of the chosen technology platform, and 2) to elaborate requirements by providing a detailed user interface of the most important functionality. The product was modelled in UML.

The first construction increment (or prototype) in the project was finished in March 2000. This increment served as an architecture release and as a requirements specification. In October/November the contractor asked the customer to provide end-users in order to evaluate the system iteratively. However, these end-users were very important engineering resources within the customer organisation, and there was considerable debate within different groups in the customer organisation regarding

<b>Project Name</b>	<b>TelMont</b>	<b>BibTid</b>	<b>UniBase</b>
<b>Description</b>	Support system used by a telecommunication company. Performs compression and optimisation activities for ISDN telephony hardware.	Promotion of journals to libraries by a Norwegian Library Organization. The application supports search for magazines, and ordering and cancellations of subscriptions.	To generate economic and administrative reports from several databases in Universities and Colleges in Sweden.
<b>#Developers</b>	10-15, plus some external consultants	3	4
<b>Contractor</b>	External, paid per-hour	External, fixed price	External, paid per-hour
<b>Duration</b>	9 months	6 months	5 months
<b>Effort Estimation</b>	Exceeded effort estimates with 100%	Effort estimates accurate	Effort estimates accurate
<b>Technology</b>	MTS/COM, C++, Oracle database, SQL	ASP/VB, Microsoft IIS, ADO, SQL	PHP, Mimer ODBC driver, SQL
<b>Process Model</b>	Evolutionary, "the Solution Delivery Process"	Evolutionary, the Genova Process	Evolutionary, Undefined
<b>Customer involved in development?</b>	Yes	Yes	Frequently asked, but did rarely answer questions
<b>Users involved in development?</b>	Yes	Yes (User = customer)	No

Tabell A.2: Overview of the studied development projects



how and when to assign these resources to the TelMont project. Finally, in early February, end-users were assigned to the project.

The part of the prototype intended to evaluate the architecture and technology became very large, and took considerably much more time and effort than planned. Based on the workshops with the end-users, the developers discovered that there was too much complexity in the initial requirement specification. A considerable amount of rework occurred as a result of the end-user feedback.

Both customer and end-user had access to the evolving system at all times. Thus, users could test solutions and suggest changes continuously. However, the workshops and the continuous evaluation of the system by the end-users produced a large number of change requests that were not always handled through formal channels.

#### *March 2000 (Changed Process)*

From March 2000 the evolutionary process (called the “Solution Delivery Process”) was abandoned and replaced by a formal waterfall process, requiring formal acceptance of detailed specifications and code at the method level. This radical change in the project was determined necessary to get economy and time schedules under control. The contractor could no longer defend using the solution delivery process because of the delays and the complaints voiced by the customer caused by the continuously changing requirements. The customer felt they lost control because the requirements changed continuously, without the formal documentation being changed accordingly. However, the developers strongly believe that the changes requested by the end-users were crucial in order to achieve a realistic and useful product.

#### *May 2000*

A detailed system specification was accepted.

#### *June 2000 (Transition)*

The first main release was in June 2000, and consisted of a fully operational system. At this time, the project had used 8600 person-hours, whereas the initial estimate was 4300 person-hours. The release was delayed by three months.

### **A.3.2 LibTime**

LibTime is an application developed for the promotion of journals to libraries by a Norwegian library organisation. The application supports search for journals, ordering and cancellations of subscriptions. The service of ordering subscriptions has earlier been conducted manually via telephone or fax. The goal of the project was to provide a web interface as an extension of this service.

The application was developed by three developers in cooperation with two end-users. The end-users were librarians employed in the customer organization. The developers and customer attended three meetings during the development. At the meetings the product developed so far was showed to the customer. The feedback from the end-users resulted in new change requests.

Initially, the requirements specification was made by the two end-users in cooperation with a consultant firm for several years, using an evolutionary approach. Originally, the consultant firm was supposed to develop the whole application, but the time limits set was delayed by more than 100%, resulting in that the customer

decided to change contractor. The requirements specification had then been going through several iterations, and many aspects of the upcoming product were discussed. The requirements specification had therefore been well thought through before delivered to the developers at Genera.

The product development did, probably as a result of rather mature requirements, not suffer from serious changes at the architectural level during the development at Genera. However, there were still a large number of continuous change requests from the end-users regarding details of the product. Change requests were also raised on parts of the product that already had been changed several times, because the end-users found it difficult to foresee the consequences of the change requests. The developers complained about too much time being wasted on discussions with the end-users regarding details of the change requests. The end-users did, on the other hand, find these discussions fruitful, and thought that the product would not have been adjusted to their real needs in the same way without these discussions. They are very pleased with the delivered product. Despite the number of end-user change requests, the software cost estimate was accurate for this project.

### A.3.3 UniBase

The UniBase system is an administrative system that has access to several economic and administrative databases in Universities and Colleges. Based on user input, the system generates reports that fit the queried data from the databases via a PHP-script. The generated reports can be read in MS Excel or a similar application. The purpose of the project was to convert the interface of a traditional application into a web interface. The reason why the customer chose to transfer the traditional application into a web-application was to make it more accessible. Having the system in on the web implies that it is easy to distribute because the web interface eliminates the need for installing the application at every user's PC.

There was no explicit requirements specification. Instead, the contractor was given the traditional application and was asked to make the web interface as similar to this application as possible.

Four developers developed the application during a five-month period. The customer was involved in making decisions when the new solution would differ severely from the original application. In these cases the communication would at times be slow.

When the product was delivered for installation, the contractor did not hear from the customer until another two months had passed. By that time, the developers had already been allocated to other development projects, and since the contractual support period had passed, all assistance and fault correction were delayed.

The UniBase project did not follow any defined process model. This was partly due to the sparse contact with the customer, and that the "requirements specification", i.e., the traditional system that was to be transferred, was well known. Furthermore, the need for a defined process was also reduced because the project was small in size and duration. The overall cost estimates were accurate, and the product was initially delivered on the negotiated time.

## A.4 Process Guidelines

This section describes recommendations based on the experiences from the three development projects. Based on these simple recommendations, we believe several of the problems experienced on the studied projects may be avoided on future evolutionary development projects of web-applications. Some readers may find the recommendations simplistic or perhaps rather obvious. Nevertheless, the actual project experiences suggest the need to reiterate the importance of recommendations such as the ones proposed in the following subsections.

### A.4.1 Project Establishment and Planning

*When estimating project costs for web development projects, be sure to account for the use of immature technology and the competency building that often is required.*

- Experiences from all the projects showed that building web applications required a considerable amount of new technical knowledge. This must be accounted for when estimating project costs.
- Web development technology was experienced to be immature, time-consuming and error-prone. Consequently, it may be harder to use than development technologies for more traditional development projects.

*Ensure that the anticipated functional and non-functional requirements of the application are supported by the chosen web development tools.*

- Experiences from all the projects showed that, for web development tools, important functionality is less refined or absent compared with traditional (i.e., more operating-system dependent) development tools. Thus, the development tools should be assessed carefully to ensure that they enable adequate support for the anticipated functional and non-functional system requirements.

### A.4.2 User Participation

*Aim to get in contact with the end-users as soon as possible. Considerable cost savings can be achieved if the end-users are involved in the project from the inception. The end-users can contribute to and evaluate the requirements specification and eliminate unnecessary functionality before it is implemented.*

- In the two projects with end-user participation, both customer and contractor found that the product had a higher quality at delivery as a direct result of the end-user participation. They developed the right product — a product that was applauded by the end-users.
- All of the users think that their participation was useful, and resulted in a product better adjusted to their work processes. The end-users thought that their presence in the project was necessary to ensure implementation of important requirements. The user engagement also increased their willingness to use the product after delivery, and to teach it to other end-users.
- User participation may result in more satisfied users. The end-users reported more satisfaction with the project they had been involved in, than with other similar projects without user participation. The end-users satisfaction was

partly due to that they had been given the opportunity to raise their opinion, and to refuse functionality that was regarded unnecessary. They also gave important input on forgotten topics and work processes that was necessary to make a well-functional product.

- In one of the projects, much time was wasted on a too complex requirements specification. When the end-users were contacted to evaluate the requirements specification, much of the functionality was considered unnecessary and was eventually removed. The customer, developers and end-users reported that costs and time would have been reduced if the end-users had been actively consulted from the inception of the project.
- Reducing complex functionality does not only reduce costs — it also adapts the product so that it fits better to what the user really needs. In one particular case, the less complex product implied that it became easier to use and easier to maintain.

*Do not expect instant feedback from end-users. Give them time to get familiar with the (incremental) product deliveries.*

- The end-users requested changes only after some period of development, that is, when the end-users were more familiar to the product and could therefore see areas of improvement more clearly than in the beginning. The contractor should be aware of that the end-users in most cases already have a full workload of regular work tasks. Consequently, the contractor should negotiate with the managers of the end-users to help them be relieved of some of their regular work tasks.
- In the studied projects, the participating end-users were not relieved of their regular work tasks. Consequently, the participation in the project was exhausting for the end-users. End-users from both projects reported that they had wished to be relieved of their regular work tasks one or two weeks to pay full attention to the development project. The contractor (and customer) should be aware of this, and negotiate with the managers of the end-users to help them to be relieved of some of their regular work tasks.

### A.4.3 Prototyping

*Focus early prototyping activities on technology evaluations. Clearly separate technology prototypes from the requirements prototypes.*

- In one of the projects, too much effort was put into a combined technology/requirements prototype. By the time the prototype was operational, it had become very large. Because of the large amount of effort spent on the prototype, it was decided to base the actual product on the prototype. This decision were taken because it was economically impossible to defend why the chosen technology should be replaced, even though the prototype proved that the technology (Microsoft/Oracle) had severe difficulties around integration issues. The technology was also shown to be error-prone with obvious limitations. We therefore recommend making a simple technology prototype during the inception of the project to test the chosen technology, and separate this prototype from other kinds of later prototyping activities. End-users are

potentially highly skilled evaluators of product functionality. However, the end-users may not be technically skilled, and consequently may have some difficulty understanding technical details before they can be demonstrated in practical terms. This gap in skills between developers and end-users may limit communication efficiency. Use prototyping as a way to bridge the gap in skill level between the end-users and the developers.

- The users in two projects reported problems with foreseeing the consequences of their choices regarding technical details. They could thereby change their mind when the change request was implemented and the results were reviewed. This was annoying and confusing for the developers. Developers should be aware of this, and for complex change requests it may be advisable to make a tiny prototype and let the end-user evaluate it before putting too much effort into the implementation of the change request.
- The users in two of the projects described it as difficult to imagine the resulting product when they reviewed the requirements specification. It was easier to evaluate the requirements specification as the development had been going on for some time, and they saw some parts of the coming product. It was first after some time they could envision the final product, and thereby got the courage to suggest changes.
- A project leader for one of the customers describes the prototypes as an essential tool for recognising requirements and visualising the upcoming product to the end-users. Without the end-users' review of the prototypes, much unnecessary functionality would have been implemented.

#### A.4.4 Change Management

*Better communication regarding software change management may increase mutual agreement around revised costs and time schedules. We recommend that a formal change management process is agreed upon and followed by the customer and the contractor. To ensure process conformance, consider using a change management tool to support the communication between end-users, customer and contractor.*

- In evolutionary development, problems with cost control as a result of change requests from the end-users may arise. While new changes are suggested by the end-users, thereby adjusting the product to better fit their needs, the customer may not see these changes as merely advantageous as the costs and time schedules may get out of control.
- Two projects experienced severe communication problems and considerable frustration as a result of not agreeing on a formal change management process. All project members reported a need for some kind of formalized process to keep track of the changes. The developers in one project were confused because the customer and end-users regularly changed their mind, and many parts of the product had to be changed several times. In the second project, the customer was very displeased about not being consulted about whether certain change requests were acceptable from an economical point of view. Both problems may have been avoided had a more structured change management process been followed. The process should formalize communication paths between parties (contractor project management, developers, customer, and end-user) to get change requests and costs under control.

- In one of the per-hour-based projects, the customer was very frustrated with the frequent changes to the requirements specification, because they caused project delays and extra costs. The continuously changing requirements specification was a source of discussion. The project manager at the customer felt that he lost control because the requirements changed continuously. However, the developers strongly believed that the changes requested by the end-users were necessary in order to adjust the product to their end-users needs. The result in the end was a cost discussion. The discussion did not get a closure until the contractor discarded several hours of work. A more formal change management process may have alleviated some of the experienced problems.

## A.5 Related Work

The results presented in this paper mostly confirm the results from related empirical studies, although there are some differences as well. The study conducted by (Emam et al. 1996) suggest that user participation is advantageous in system development with uncertain requirements. When uncertainty is low, the beneficial effects of user participation diminish. On such projects, user participation could instead be annoying to the end users because they feel their efforts are redundant. This resentment may bring out reductions in quality of service as user participation increases. These results are in accordance with our observations of the successful user participation in TelMont and LibTime versus the non-participative approach of the UniBase system. In the UniBase system, the end users were unwilling to contribute to the development. The requirements specification was from their view given with no uncertainties, that is, they wanted their traditional system on web with no changes in functionality or design. They seemed very satisfied with the final product despite their sparse contact with the developers through the development process. In the two other projects, however, the users did not know exactly what requirements and functionality the final product should include. The requirements specifications were refined and changed several times by prototyping their suggestions and through dialogues with the developers. The end users found that their participation in the system development were necessary and useful in developing a right product that matched their needs, because they were unable to see what they really wanted before parts of the product were prototyped. Their presence also reduced the developers' uncertainty regarding the requirements specification.

The study conducted by (Zamperoni et al. 1995) indicate that sometimes users or customers had a specific vision of the future system, but are unable to formulate this perception properly to the developers. This problem could be improved by using evolving prototypes. The prototypes could be useful in communication — in being a mediator to reach agreement with the users and their expectations. In accordance with our results, Zamperoni et al. also found that the product is more likely to be accepted among the users if they are involved in the development process. However, they stress that short communication channels between users and the development team are important prerequisites to increase the likelihood of acceptance. Similar results have also been reported by (Ehn 1993). He argues that the end users are highly qualified in evaluating the product functionality, but that they do not necessarily have the technical skills to understand the requirements before it is demonstrated in practical use. Ehn points out that this gap in technical

skills may be bridged by the use of prototypes. We made the same observation in our study, as the users pointed out that it was easier to contribute after a while, when they were able to see some parts of the final product, not just a technical specification.

The study in (Lichter et al. 1994) presents a critical view of end user involvement. They found that developers may do the mistake to encourage the end users to voice all ideas and wishes that come into their mind when they evaluate a prototype. This may lead to the incorporation of every conceivable function or design option, and thus increases the complexity and reduce the usability of the final system. This was not the case in our study. Especially in the TelMont project, we rather found that the user participation reduced the complexity of the requirements specification and the final product.

On the TelMont project, user participation resulted in a less complicated requirements specification and made the product both more maintainable and easier to use. However, as Bratteteig (Bratteteig 1997) suggests, reducing functionality in this way may not be an easy decision to make for the developers. The end users' needs of having the easiest possible product to use may collide with the developers wishes of making a product that incorporates new technology and impressing functionality that is more challenging for them to code. To the end users, technical elegance may not be as important as it is to the developers. On the TelMont project, this tension was experienced, but was solved smoothly. The management wanted a product that incorporated many technical details and several ways of navigating. The end users, on the other hand, wanted the system as simple as possible, with one restricted way to navigate through the product. It is a challenge to the developers to accept this, but in TelMont they chose to go for the end users solution, and the result was highly satisfied users.

Hohmann (Hohmann 1997) argues that the design of the developer organization's processes is reflected in the design of the products they make. He reasons that if the developer organisation has well organised communication channels and is properly structured, it will be reflected in the design of their products. A well-defined organisational structure constitutes a shared belief amongst the developers of how the job should be performed, and this may in turn lead to that the product is well structured. A parallel can be drawn to formal change management, e.g., as proposed in this paper. A good structure and well-defined communication paths for the change requests may structure the change management, and avoid that it comes out of control.

Forsyth (Forsyth 1998) shows by empirical material that planning a strategy to use in development of products (e.g. process improvement) contributes to better verbal communication, increased motivation and higher flexibility in conducting the planned tasks. If we draw the parallel to change management, planning the change management process may be advantageous to both communication around - and structuring of - the change requests so that they do not run out of control.

## A.6 Conclusions and Future Work

This paper described current efforts to develop the Genova Web Process, tailored to support evolutionary development of web applications. Based on experiences

collected from three web development projects, specific process guidelines on important topics such as initiation, planning, user participation, prototyping and change management were proposed. We believe these guidelines may be useful for other, similar types of projects.

### **A.6.1 Future Work**

One of the guidelines suggests a need for a better way of handling change requests in evolutionary development of web applications. Uncontrolled changes had been a source of frustration in the projects. As a response to these observations, a simple but formal change management process is under development. A tool (the “change management prototype”) that supports this process is being implemented.

At present, the change management prototype will be evaluated on a new web development project in which Genera is the contractor. During the project, the prototype will be used by the customer, end-users and developers. Interviews will be conducted to evaluate the effect of the formalised change process tool, and to determine whether it should be refined and developed further into an integrated, commercial part of the Genova Web Process.

### **A.6.2 Acknowledgements**

We thank the interviewed participants of the three development projects for spending valuable time and for allowing us access to potentially sensitive information. The research project is financed by The Research Council of Norway through the industry-project PROFIT.



# Tillegg B

## Intervjuguide

Dette vedlegget inneholder intervjuguiden med åpne spørsmål som ble brukt under intervjuene med kunde- og utviklerorganisasjoner.

### Generelt om prosjektet

- Hva var formålet med prosjektet?
- Hvilke komponenter består systemet av?
- Var det flere forskjellige underleverandører som deltok i systemutviklingen?
  - Hvis ja: hvordan samkjørte systemene deres under test?
  - Var det noen problemer med å ha flere underleverandører? Hadde underleverandørene separate avtaler?
  - Greide de å samarbeide med hverandre på en tilfredsstillende måte?
- Når startet prosjektet?
- Hvordan ble initiativet til prosjektetableringen tatt?
- Var det fastprisoppdrag eller timebasert?
- Var det hard konkurranse med andre bedrifter om å få oppdraget? Påvirket det prosjektetableringen eller prosjektgjennomføringen negativt?
- Var dere en intern eller ekstern leverandør?
- Hvilke firmaer har deltatt i systemutviklingen?
- Hvilke deler av prosjektet hadde de forskjellige firmaene ansvar for?
- Hvor mange utviklere var dere i prosjektet?
- Hvem skal vedlikeholde systemet i etterkant?

### Roller

- Hvilken rolle hadde du i prosjektet?
- Trengte du mye ny kompetanse for å løse arbeidsoppgavene dine?
- Hvordan tilegnet du deg ny kunnskap for å løse oppgavene dine dersom dette var nødvendig?
- I hvor stor grad følte du at du samarbeidet med andre utviklere om prosjektet?
- Var det noen roller du synes var overflødige for prosjektet?
- Var det noen roller du savnet i prosjektet?

## Kravspesifikasjon

- Hvem lagde den initielle kravspesifikasjonen? Kunden, utviklerne eller var det et samarbeid?
- Brukte dere lang tid på å lage den initielle kravspesifikasjonen? Hvor lang tid i så fall?
- Var brukerne med på å lage den initielle kravspesifikasjonen? Evaluerte de den?
- Var i så fall brukerne nyttige for å endre på noen av kravene? Kom de med forslag til hvordan produktet kunne bli annerledes?
- Reviderte dere kravspesifikasjonen mye underveis? Hvor mange ganger reviderte dere den? Eller var det mer en kontinuerlig forandring på kravspesifikasjonen?

## Brukermedvirkning

- Deltok brukerne i utviklingen av produktet?
- Hvordan fant dere fram til ulike brukergrupper?
  - Var dette en vanskelig prosess?
  - Hvordan var kommunikasjonen mellom de forskjellige brukergruppene?
  - Hvordan satte dere sammen en bra referansegruppe (hvis dere gjorde det)?
- Var brukerne nyttige for å skape produktet, eller ville dere klart å lage produktet uten deres hjelp?
- Kom brukerne med endringsønsker underveis i prosjektet?
  - Hvis ja: prøvde dere å følge opp disse endringsønskene?
  - Ble disse endringsønskene først foreslått for kunden før dere implementerte dem?
- Beregnet dere en pris på endringsønskene?
- Var dere bevist på å fortelle brukerne hvor mye disse endringene ville koste, eller overlot dere til kunden å avgjøre ønsket ut i fra en eventuell pris dere satte opp på hvor mye endringsønsket ville koste?
- Kom det så mange forslag til endringsønsker at dere ikke kunne følge opp alle endringene?

## Prototyping

- Lagde dere én prototype, flere små eller ingen prototype?
- Var prototypingen viktig?
- Hva brukte dere prototypen til?
- Utviklet dere prototyper for kunden, eller hadde dere en klar avtale om hvordan produktet skulle se ut før utviklingen startet?
- Lagde dere en prototype der dere presenterte endringsforslag for brukeren? Hvordan fungerte det?
- Er det noen ting ved prototypingen du synes fungerte bra? Dårlig?
- Hvordan ville du prototypet dersom du skulle gjort prosjektet på nytt?

## Kunderelasjon

- Hvor god kontakt følte du at dere hadde med kunden?
- Hvor ofte fikk dere feedback fra kunden?
- Var dere i liten eller stor grad i kontakt med kunden på begynnelsen av prosjektet? Underveis i prosjektet? Etter prosjektet var avsluttet?
- I hvor sterk grad var kunden villig til å samarbeide med dere?

## Prosessmodell

- Hvilken type prosessmodell kjørte dere på prosjektet?
- Var dette en prosessmodell dere var godt kjent med fra før, eller var det en prosessmodell ingen av dere var vant til å bruke?
- Hvordan synes du denne prosessmodellen fungerte?
- Hva fungerte bra ved prosessmodellen? Dårlig?
- Ville du brukt denne prosessmodellen om igjen dersom du hadde skulle kjørt prosjektet på nytt?
- Hvilken prosessmodell tror du i så fall kunne vært bedre egnet?
- Hvilken type prosessmodell ville du selv valgt dersom du hadde muligheten til å bestemme over det?
- Hva ville du gjort annerledes dersom du skulle gjøre prosjektet om igjen? Er det spesielle punkter i prosessen du ville utelatt, forandret på eller brukt mer tid på?
- Hvor mange inkremitter hadde dere i løpet av utviklingsperioden?
- Synes du dette var for mange eller for få inkremitter?
- Ville dere hatt færre/flere inkremitter dersom dere skulle utført prosjektet på nytt?
- Evaluerte kunden inkrementene?

## Aktiviteter

- Var det noen delaktiviteter du synes dere brukte for mye tid på? For lite tid på?
- I hvor stor grad dokumenterte dere arbeidet deres?
- Var det noen delaktiviteter du savnet som en del av prosjektutviklingen?
- Hvor mange av prosjektdeltakerne deltok i designfasen?
- Var det de samme menneskene som designet og som kodet systemet?

## Layout

- I hvor stor grad vektla dere layout og visuelt design i prosjektet?
- I hvor stor grad vektla dere å gjøre applikasjonen konsistent layoutmessig? (At sidene hadde en konsistent layout etc.)
- Investerte dere mye tid og ressurser på å gjøre layouten til applikasjonen innbydende?
- Gjenbrukte dere komponenter fra andre webutviklingsprosjekter?
- Lagde dere et navigasjonskart (kart som forteller hvordan brukeren kan navigere seg mellom de ulike sidene)?
- Var brukerne med på å utforme layouten? På å tegne navigasjonskart/flytdiagram?

## Om webutviklingsprosjekter

- Har du vært med på mange prosjekter der dere utvikler webapplikasjoner?
- Hva synes du er forskjellene mellom å utvikle webapplikasjoner og tradisjonelle applikasjoner?
- Hva er fordelene med å utvikle for web? Ulempene?

## Testing/feil

- Er designet dokumentert i UML eller annen tilsvarende notasjon? Hele eller deler av designet? Er modellene holdt oppdatert med nye leveranser ?
- Hvilke deler/komponenter av koden brukte dere mest tid på å rette opp feil på?
- Har dere brukt mye tid på å rett opp feil i kravspesifikasjon? Design? Kode? Måtte dere redesigne noe av systemet etter kodingen var startet og dere hadde fått sett hvordan det ville bli i praksis?
- Fant dere kritiske feil under testingen? Hva tror du disse skyldes?

## Estimater

- Klarte prosjektet å holde seg til det opprinnelige estimatet, eller ble det estimatoverskridelser?
- Hva tror du estimatoverskridelsene skyldtes?
- Har du noen forslag til hvordan noen av estimatoverskridelsene kunne vært unngått?

## Tillegg C

# Transkribert materiale

Vedlagt følger de fire viktigste intervjuene transkribert ordrett. Materialet er fra intervjuer fra TelMont-prosjektet, og intervjuene er med henholdsvis prosjektleder hos kunde, samt prosjektleder og to utviklere hos utviklerorganisasjonen. Forkortelsene E og A i intervjuene står for henholdsvis Erik Arisholm og Anette Lien, som var de to som foretok intervjuene, mens forkortelsene S, H, B og K er initialene til respondentene. Det som står i parenteser er forklaringer gjort da jeg transkriberte materialet, for eksempel der jeg husket at en av respondentene reiste seg og tegnet på en tavle. Der det står “(uklart)” har det vært uklart opptak på lydbåndet. Organisasjonsnavn er anonymisert og dermed det eneste som ikke er skrevet ordrett fra tape. Her er kundeorganisasjon kalt “organisasjon A” mens utviklerorganisasjon er kalt “organisasjon B”.

### C.1 Intervju med prosjektleder hos kunden på Tel-Mont

A: Men det første spørsmålet: Har du noen erfaring med å utvikle webapplikasjoner?

S: webapplikasjoner er på en måte veldig godt kjent i organisasjon A. Vi har funnet ut at det er mye vi trenger å utsette for web, for å si det sånn...

A: Mmm

S: og internt i butikken, så er det veldig mye som er gjort på web allerede. Så det finnes webting som vi har som er laga fra før.

A: Mmm. Okei.

S: for at vi har umulig antall grensesnitt der folk kan jobbe fra hvor som helst, derfor er web en veldig fin ting, så det optimaliserer organisasjonen, da.

A: Mmm. Okei. Synes du andre webutviklingsprosjekter du har vært på har fungert annerledes, eller bedre eller dårligere eller...

S: Siden det her var mitt første oppdrag som prosjektleder, må jeg si at jeg synes andre prosjekter har vært bedre drevet. (alle ler) Men...men...de andre prosjektene jeg har vært på har enten vært veldig små, eller så har...en veldig liten gruppe, uten forretningsmessige avtaler, eller så har det vært noe som har vært veldig stort, men...der man på en måte har jobba i en enhet som har sine egne mål, så det her er første gangen for meg at jeg har vært borte i så mange del-leveranser. Og det kunne vært gjort mye bedre.

A: Ja, okei. Det kunne vært gjort bedre?  
S: Ja. Fordi at dette her var jo tross alt første gangen.  
A: Ja, nettopp. Men har du noen forslag til hvordan du ville gjort det bedre?  
S: Jeg ville forankra det bedre i ledelsen, og få gjort de rette avtalene først, og funnet ut - hvem er min medhjelper i butikken. Og organisasjon A er kjempesvært, det er så svært at.. Da er det gjerne sånn at da bruker du de midlene som er til å se hva er det som finnes...(uklart)  
E: Når du sier "forankre bedre i ledelsen", på hvilken måte?  
S: I og med at det finnes nedfelt i politikken til forskjellige enheter at man burde bistå sånn og sånn og da og da, så må man kanskje planlegge en sikker måte, eller en måte å sikre seg på at man har de midlene, fordi at..en så stor butikk som det her, da, da er du nødt til å skrike litt for å få oppmerksomhet, da.  
A: Mmm.  
S: Altså, om mitt prosjekt går helt ad undas, er det ingen som merker seg det, så på den måten så blir det bare en rød strek i boka, og så...går de videre, ikke sant..  
E: Sa det var på en måte problemer med å få fokus på det prosjektet her i...  
S: Helt i startfasen, ja. Men det gikk annerledes når vi hadde del-leveranse. Trikset var del-leveranser. Og skal vi se på forholdet til de som hadde utviklinga i TelMont, så var ikke ting, altså, de klare målene var ikke definert godt nok. Vi brukte de til en analysefase som nesten umerkelig gikk over i en utvikling. Og det var lite ...lite kommunikasjon om: nå har vi passert der, nå har vi passert der..så time for time, det kommer jeg aldri til å gjøre mer. For utvikling, i hvert fall....  
A: Det fungerte dårlig, ja. Synes du det har vært mange likhetspunkter mellom TelMont og andre webutviklingsprosjekter du har vært på?  
S: Nå har jeg faktisk ikke personlig erfaring så god at jeg kan trekke de helt klare forskjellene. Men på rykter, så høres det ut som at vårt forhold til organisasjon B er prikklikt alle forhold vi har til organisasjon B. Så det er..ja, det er kaos og bråk om...i det forholdet der.  
A: Å ja. Mmm. Eh...Har du høstet noen erfaringer du synes er viktige for å lage sånne generelle retningslinjer om hvordan man skal gå fram når man utvikler webapplikasjoner?  
S: For webutvikling?  
E: Spesielt, eller generelt.  
S: Ja, la oss ta generelt først, kanskje.  
A: Ja.  
S: Eh....Altså, det finnes en driftsleverandør, og det finnes en sånn programvareleverandør. Og ha litt fokus på at de her samarbeide littegranne. Og hvis det er flere underleverandører, så er det for svakt å ha en separat avtale med den og en separat avtale med den uten at de i det hele tatt har noen felles forpliktelser. Og her var det i grunnen tre miljøer i organisasjon B som skulle gjøre forskjellige ting, og de...de nekta rett og slett å underordne seg det ene eller det andre miljøet. De kjenner hverandre selvfølgelig veldig godt, men de tar ikke noe ansvar for hvor naboen deres er, det...altså, det ble en snedig konstellasjon, så...(uklart)  
A: Og konkret for webutviklingsprosjekter?  
S: For webutvikling da er det utrolig mye som kan utsettes for et webgrensesnitt, noe er fancy og noe er fint og noe er det kanskje like greit og latt være. Og...og...det er...Du legger utrolig mye føringer allerede i en analysefase fordi at...Altså, det her om man opplever med web, er at businesslogikken som ligger bak kanskje kan være litt, så....Men når man spesifiserer ting for...for eksempel web, da konsentrerer man

seg mange ganger om skjermbilder, og når man skal gå i gang å lage..at prosjektet skrider fram, så bør man huske at de skjermbildene som man har i en prototype ikke er spesifisering, de er de man utvikler på. Så vidt jeg har forstått, så må nesten en utvikler få lov å utvikle littegranne for å tenke. Han tenker mens han utvikler, ikke sant.

A: Mm.

S: så dermed når de lager detaljspesifikasjonene sine, så har de i grunnen masse ferdig på skjermbildene sine. Og, og da er det vel egentlig to ting som er litt viktig, og det første er at det at de har utvikla mens de tenkte ikke nødvendigvis er sånn som vi har nødt til å godta. Og det andre er at når de...de forholdt seg mot våre brukere, altså, prosjektet var organisert med brukere som fikk se på skjermbilder og se at...ja, det var fint og det var fint...

A: Mmm..

S: De framtidige brukerne, og på det her prosjektet som var time for time, så var det veldig lett å si at: å, den var fin, den vil vi ha, og så sier utviklerne: “du, jeg tror det ville vært lurere sånn og sånn”, ikke sant, og så setter dem det opp på skjermbilder, og så er man fornøyd med det, og så viste det seg jo at det var en helsikes businesslogikk som måtte utvikles bakom det her, på grunn av en liten ting, så, små endringer, kanskje bare å føre opp en ekstra dropdown-meny her med forskjellige..med sorteringsmekanismer er kjempesvært bakom. Og det her kan drive litt ut av kontroll. Hvis man ikke er helt soleklar på: hva er det som er spesifiseringen, hva er det vi har bedt om i grunnen.

A: Ja.

S: Og...Ja, utviklerne de bør kanskje da også vite at de..de fører på noe nytt da. At det her koster penger, og at du må fortelle nesten brukerne at du tilbyr dem noe som ser enkelt ut, men som er dyrt. Sånn at prosjektlederen som sitter med pengene veit hva som foregår.

E: Synes du det var problem her?

S: Det var et problem.

E: At de ikke sa ifra om det?

S: Eh, de regna vel med at våres brukere sa ifra. På en måte, så at våres brukere regna med at dette her ikke var så forbaska dyrt. Og..og..ja, med to uerfarne prosjektledere i toppen, så, så var ikke det så greit å holde styring på. Det var ikke bra nok kommunikasjon. Fra bunn til topp hos organisasjon B og de tingene de forstod, og det var kanskje ikke bra nok kommunikasjon fra bunn til topp på de tingene vi forstod. Men de forstod jo definitivt mye mer av applikasjonen og av utviklinga enn vi forstod. Vi forstod bare: hva er det det her kan brukes til, og hvor gjevt det her kommer til å bli den dagen det er ferdig. I og med at brukerne kanskje ikke hadde så mye teft for utvikling, som man selvfølgelig ikke kan forlange at de har, så var det overskridelser til å begynne med. Men nå på den runden vi har nå, så har det blitt veldig bra kontroll, veldig nøye vurdert. Og vi jobber veldig bra, de her to butikkene våre, veldig bra i lag.

E: Men jeg lurer på — når du sier det da, så lurer jeg på, det høres ut om mye av det her henger sammen med hvordan man på en måte håndterer endringer.

S: Endringsstyring, ja. På et time-for-time-prosjekt, så er det ikke fokus på endringsstyring. Hvertfall ikke hos leverandør, der er det bare: “ja, ja, ja, her er det bare å ture på”, men den store feilen som ble gjort den her gangen, var at vi fikk ikke tilbakemelding før det hadde gått to og en halv måned. Vi fikk ikke rapport om forbrukte timer, vi fikk kun en faktura, den fikk vi i januar, og da var en del

aktiviteter allerede sprenget med. (Uklart)men på den måten var det mye som var låst allerede, uten at vi i grunnen visste om det.

E: Men det er da jeg lurer på med den endringshåndteringen, det er da du sier at nå blir det mye bedre. Men da lurer jeg på: har dere gjort konkrete tiltak for å...

S: Ja, en fastpriskontrakt det er liksom første steget. For har du en fastpris, da får du fokus, få en designreview, og bli enige om at..La oss si at vi har kravspek først, detaljspek, som kan gå på en slags designreview, det er noe vi ikke valgt å ta med den her gangen, det er en omforent kravspesifikasjon, det vil si at vi godkjenner den, og dermed er berettiga til en garantiperiode etter leveransen. Men vi ville ha så og så mye kroner for å få det her til å bli omforent, så å få godkjent at detaljspesifikasjonene ser ut som at de oppfyller det som står i våre kravspesifikasjoner, og så utvikle på grunnlag av det, og kommer det endring da, så ser man at en endring i detaljspesifikasjonene, da har produsent bedre mulighet til å sette en pris på det her, og det går an å ta opp endringer, og si noe om det er mulig å levere innenfor de gitte rammene. Hvertfall ikke kravspekk uten at du har full kontroll på hva du gjør. For det var hele tida det...første trinnet var. Der virka det som om enkelte ting gikk på endring av kravspekk, og det er lett å miste kontroll, rett og slett. Du skal være veldig god for å ro det inn. Så samarbeidsmåten som vi hadde da, var mer eller mindre sånn at man utvikla samtidig som man analyserte. Og kontrollere pengeflyten der, det er litt vanskelig. Man visste at vi hadde vært, altså, nå kommer noe som er litt interessant. Vi var, når vi begynte prosjektet., så var organisasjon B en del av organisasjon A, da var det policy at det her skulle gjøres sånn. Så bryter organisasjon B ut av organisasjon A, da er det en policy at det ikke skal gjøres sånn, men vi er midt oppe i det. Hva gjør vi da? Da får vi et lite dilemma, plutselig går vi over til en organisasjonsstruktur som sier at det her burde ha vært fastpris.

A: Mmm.

S: Så det var en utfordring her, da.

A: Ja. Jeg hadde tenk å spørre mer om den kravspesifikasjonen etter hvert, da.

S: Men for å utfylle spørsmålet, så er det litt om test også da. Altså, en test foregår jo som en integrasjonstest, blant annet, altså, modultest først, integrasjonstest, det her foregår hundre prosent hos leverandør. Så kommer systemtest, det er deres ansvar. Så kommer akseptansetest, det er vårt ansvar. Den testen som var gjennomført her, var at vi hadde systemtest og akseptansetest i lag, vi brukte felles ressurser. Og organisasjon B hadde en oppfatning av at det skulle testes veldig. Altså, vi hadde forskjellige oppfatninger av hvor dypt vi skulle teste. Det viste seg jo at vi fikk jo aldri systemene til å virke under systemtest, vi systemtesta og systemtesta, og vi hadde underleveranser som ikke makta det de hadde lovt. Og da fungerte jo selvfølgelig ikke systemene i lag, og vi måtte på nytt og på nytt og pånytt. Vi fikk aldri kjørt hele verdikjeden, før en måned etter at vi skulle ha vært ferdig. De to andre underleverandørene hos organisasjon B har to forskjellige organisasjonsmåter, den ene er en veldig god prosjektorganisasjon. De hadde mase problemer selvfølgelig, fordi vi hadde bedt om noe avansert, og da kan man jo akseptere at det kommer feil og sånn, den andre har ikke noe god prosjektorganisasjon, men de er jævlig gode folk. De jobba natta og dagen for å få til, men allikevel gikk det ikke helt når de kom med ferdig vare. Først leverte vi kravspekken, så gikk det en lang stund, og så kom de og sier at de hadde endra kravspekken vår, og så greide de ikke levere det som de skulle. Og det skjer, leveransen fra dem skjer mens vi systemtester, og leveransen feiler hele tida, derfor



får vi ikke ferdig systemtesten.

E: Hvem snakker du om nå egentlig?

S: Okei, jeg snakker om tre leverandører under organisasjon B.

E: Ja vel.

S: to etablerte støttesystemer som skal levere rapporter til oss, så TelMont, som lagde den applikasjonen som dere har intervjuet. Det var ikke X, det var to andre miljøer i organisasjon B.

(Tegner tegning)

S: Systemtesten ble dårlig fordi det ene firmaet ikke greide å levere til tida. De kom med endringer veldig seint, og leveransen kom enda seinere. Sjøl om det er veldig gode utviklere det her, de er viden kjent for å være veldig gode, så er kanskje ikke prosjektorganiseringa deres helt optimal. Men nå skal vi ikke gi dem hundre prosent av skylda. Vi har en egen gruppe i organisasjon A, som skal holde de her i øra, så først og fremst må vi ta sjølkritikk og snakke med den gruppa først, før vi i grunnen retter noe flengende kritikk mot dem. Prosjektet her opplevde den feilen der, og det som kanskje er verdt å lære for dere, er at underleverandører er ikke helt enkelt. Og det ser man jo på bygningsprosjekter også. Og jeg burde kanskje lært meg mye mer om hvordan man håndterer underleverandører.

A: Men har du noen forslag til hvordan en kan håndtere det?

S: i og med at vi har noen som kan holde dem i øra, så burde vi forankret det her mye bedre.

A: Over til noe annet - har dere noen spesiell prosessmodell som dere pleier å bruke på.

S: Det er utvikla en prosjektmodell i organisasjon A. Som er i grunnen en sånn multipurposemodell. For at du kan plukke ut det du ikke trenger.

A: Å ja.

S: Men som uerfaren prosjektleder, så var mye av den her modellen veldig stor. Det var mange verktøy der som jeg ikke helt hadde begrep om åssen jeg skal benytte, så. Det vil si, modellen gjør at: bruke det og det, gjør sånn og sånn, men alle påbudene der var litt vanskelig å følge, den er litt rigid, og det var fristende å ta bort en del av de påbudene.

A: Ja, men gjorde du det eller?

S: Eh..på en kanskje ukontrollert måte så var de tatt ut. Men prosjektmodellen er god nok for store prosjekter, og de aller største prosjektene de kjører den prosjektmodellen der, og man er veldig fornøyd, så det er ikke det. Det går på meg da.

A: Okei. Er det en sånn typisk iterativ prosessmodell?

S: Ja, den er iterativ. Der du har beslutninger imellom. Og de store prosjektene har også plikt om å revurdere egen nedleggelse under hver eneste fase de er ferdige med.

A: Mm. Har du en prosessmodell du ville foretrukket å bruke framfor den?

S: I grunnen ikke. For nå tror jeg at jeg kjenner den bedre, og kan lettere bruke den. Så prosessmodellen hadde vært fin å bruke littegranne før jeg gikk inn som prosjektleder.

A: Mmmm.

S: men jeg har aldri vært prosjektsekretær eller noe tidligere.

A: Nei vel.

S: så på den måten så var det ikke helt...Da varden litt nye for meg, rett og slett.

A: Ja, for deg ja. Men for TelMont?

S: Nei, altså vi har en person som har erfaring fra masse prosjekter, de andre brukerne har ikke hatt noen sentrale roller i prosjekter fra før.

A: Nei. Men var det litt frustrerende at ingen hadde kjennskap til den prosessmodellen?

S: Ja! Og ingen hadde motivasjon for å følge den til punkt og prikke. Jeg har vært slitt mellom: skal jeg tilpasse meg prosjektdeltakernes ønsker, eller skal jeg tilpasse meg prosjektmodellen.

A: Mm.

S: Og skulle jeg hivd meg inn på en prosjektmodell mange oppfatta som rigid, så var det...Når vi stod i stampe, så var det litt panikkaktig, man tok da øks og hammer og..

E: H nevnte at de brukte i begynnelsen av prosjektet noe de kalte løsningsleveranseprosjektmodellen, er det den du refererer til?

S: Nei. De har en annen modell, de altså. Begrepene stemmer ikke i hop.

A: Brukte de en annen prosessmodell enn dere da, eller?

S: de brukte en annen prosjektmodell, og den rollen som vi spilte i deres prosessmodell, var ikke godt nok forklart.

A: Men var du misfornøyd med det?

S: Nei, det var veldig bra at de brukte den prosessmodellen, men vi så jo ganske tidlig at de stemte jo ikke i hop, og det var ikke gjort så enorme forsøk på å matche de før vi begynte prosjektet. Og prosjektmodellen var ikke omforent, på en måte. Før det hele starta.

A: Okei. Men prosessmodellen du brukte, var den for alle tre samtidig, liksom?

S: Altså, de her de inngikk jo i de første, altså, de var med i analyse, de var jo med i konsept- og designfase, og da konsept og design var ferdig, da hadde vi fulgt modellen var ganske bra, i hvert fall det som er felles mellom de modellene har vi fulgt ganske bra, da. Men, så plutselig, så de Y her, som liksom helt hoppa ut av sporet omtrent, og ga meg ny kravspek, lenge etterpå.

A: Altså, det er så rart at dere hadde en prosessmodell og TelMont hadde en annen?

S: altså, organisasjon A hadde en modell, organisasjon B hadde en annen modell.

A: Og så måtte dere få de til å passe sammen?

S: Ja, det var det vi ikke gjorde noe forsøk på. Vi så at det her var så lite prosjekt at det kanskje ikke var noe vits i å..Om vi hadde en merkantilt ansvarlig med helt fra starten av hos begge partene, så hadde det her helt sikkert vært satt på plass, og det er en av de tingene jeg burde bli bedre til, det er å bruke merkantilt mye mer. Og de er veldig glad for å få lov til å bli brukt og. For å si det sånn. De er en veldig god organisasjon i seg sjøl.

A: Hva er det merkantilt?

S: Altså, det er en sånn som har innkjøpserfaring og som vet hvordan det her skal foregå.

A: I organisasjon A?

S: Ja, altså de her er generelt gode i innkjøp, vi har en egen sånn organisasjon i organisasjon A, på samme måte som ethvert stort firma har en sånn intern organisasjon. organisasjon B har og sin egen. Og de brukte sin merkantile kontakt helt i starten. Og, og, han var jo mye med på at i hvert fall de brukte den modellen som organisasjon B har, da. Han er en sånn formell kontakt mellom de to organisasjonene.

A: Okei. Mm. Er du fornøyd med å bruke sånn iterativ prosessmodell, eller?

S: Ja! Det er jeg.

A: Du er det, ja. Du vil ikke bruke noen annen type prosessmodell?

S: Jeg har ikke prøvd noen særlig annen enn den iterative fossefallsmodellen som du kanskje sikter til nå.

A: Eh...iterativ prosessmodell???

S: Altså, for meg så er det samme greie, er det ikke det?

E: altså, iterativ er ikke faseorientert på samme måten, det er mer at man har analyse, design, kode, test parallelt.

S: Aaaaaaah! Akkurat. Nei, en sånn har jeg ikke vært med på, jeg kan ikke uttale meg om det. Men innenfor de fasene som vi hadde, så var det jo mye testing samtidig som produksjon. Fordi at dette her var et webprosjekt. De jobba på en prototype hele tida. De kunne ikke jobbe noen annen plass enn på en prototyp i grunnen. For å vise hva slags resultater vi hadde. For at vi har brukere som ikke forstår noenting hvis vi kommer med en frase eller et utsagn. De forstår det når vi kommer med et papir eller på skjermen, eller...og da var det att og fram, att og fram. Og utviklerne måtte selvfølgelig prøve å gjøre det de hadde blitt enige om i dag, da. For å se om det her funka før de kom tilbake igjen. Så inne i den utviklingsfasen som var i grunnen kjempelang, så kom hvertfall forsøk på en prototype, og presentasjon av siste endringer, var det her godt nok, og så prøve på nytt. Samtidig som man ble enige om ting, så skrev man testspesifikasjoner, og da skrev man testspesifikasjoner både for systemtesten, og for akseptansetesten. Men det var kun for at vi brukte de samme ressursene på de to testene. Og hadde de drevet på for seg sjøl, så hadde de måttet skrive systemtestspesifikasjoner, mens vi måtte skrevet akseptansetestspesifikasjoner. Så det ligger parallelt samtidig som du har utvikling, samtidig som du har prototyping. Et utrolig kaos kanskje for meg som prosjektleder, og det her time for time har vært ganske vilt, egentlig.

E: Men selv om du synes det ble litt kaos, så ville du.?

S: Ja, la oss nå si: hvordan ville jeg gjort det annerledes? Nå når vi kjører fastpris på det her, så er ting mer kontrollert. Da er miljøet tilstede for å kjøre en sånn iterativ del i løpet av utviklingsfasen, så det går an å gjøre det. Det er ingen som angrer seg på det nå. Så for så vidt er jeg fornøyd med den fasen som vi har gjennomført. Og trinn 1, som la grunnlaget for resten, da var det ikke kontrollert nok.

E: Du mener altså at problemet er ikke prosessmodellen, det er mer endringene.

S: Ja, og ha en endringsstyring, ja. Og kanskje en litt mer formalisert forslag til endringer. Altså, litt mer formaliserte designreview, det hadde vært fint.

A: Var det noen spesielle deler av prosessen du syntes fungerte ekstra bra eller ekstra dårlig eller.

S: Ja, altså den kommunikasjonen fra bunn til topp, for å kalle den det. Den var ikke god når det gjaldt å fortelle om kostnader. Produktet i seg sjøl er jækla bra, vi får trampeklapp hos brukerne. Og kreativiteten var veldig godt tatt vare på. Så det er en balansegang her, kanskje.

A: Ja.

S: Er du kreativ, så går det vel.(latter) altså, vi hadde ikke så voldsomt behov for kreativitet i fase 2 av det her, så dermed var det veldig lett å kjøre på en rigid endringshåndtering.

A: Men uten at det hadde vært mye endringer, så hadde dere kanskje ikke fått det bra produktet?

S: Eh...Kan...Altså, vi måtte kanskje hatt en til to store endringer, da. Og hadde vi kjørt en klar designreview på det, og gjennomført de to store endringene, for eksempel. Så hadde det her vært nesten like bra uansett.

A: Ja, nettopp.

S: For da hadde det vært bare å notere seg hva som var foreslåtte endringer, gi de status: "foreslått - ikke avklart". Det er noe som heter endringsønsker, det er et kjekt begrep, ikke sant, du ønsker at ting skal bli endra.

A: Mm. Eh...Hadde du forventet høyere produktivitet enn det ble i prosjektet?

S: jeg var forespeilla høyere produktivitet. Estimatene helt i starten var helt fullstendig. De var mye mindre enn det de. Og da at vi kjørte time for time gjorde at det var veldig lett å kanskje, dra ned littegrann. De første estimatene som kom her, de kom lenge før at alle endringsønsker var kartlagt.

A: Mmm.

S: Det er litt skummelt.

A: Mmm.

S: Og de nye estimatene kom omtrent før de kom når de så at "her var det overskridelser, da må vi ha ett nytt estimat". Et nytt estimat burde kommet før de så at det var overskridelser.

A: Men tror du det kan skyldes litt at det var såpass mye kreativ tenkning?

S: Ja, litt det, og så...Hva er det de har sagt om...

A: Nei, det var bare mye at det var endringer i kravspesifikasjonen, at brukerne ville ha endringer og....

S: Det her er det sikkert ikke OK at dere skriver, men la oss si at de..det kom et uforutsett problem med, i utviklinga deres, ting som ikke matcha helt, for å si det sånn.

E: Teknologi?

S: Teknologiproblem. Teknologivalget var ikke gjort. Valget burde ikke være gjort før man hadde utreda alle konsekvenser av valgene man hadde gjort. Men de er ikke dårlige fordi de har kommet opp i et sånt problem, bare for å få sagt det.

A: Ja.

S: Det er de ikke. For at det de ikke fikk til, det var jo matchinger mellom ASP og COM, sånn som jeg forstod det.

A & E: Mmm...

S: Og mot Oracle, de måtte jo leie inn spesialhjelp, da fikk de jo ting på plass.

A: Du var misfornøyd med det...

S: (Avbryter) Nei, jeg var ikke misfornøyd med at de kom opp i et problem, det som var skuffende var at de tok det her for god fisk på forhånd, og ga et timeestimat som vi trodde fullt og fast på, som....For det fikk jo innvirkning på business-caset.

A: Men det var liksom teknologien som var litt feil valgt, ikke sant, men skulle du ønske at dere hadde valgt teknologien for dem?

S: Nei! Men...Men det kan være at den ble valgt på et litt tynt grunnlag. Og det har de lært av. Nå gjøres ting..Neste gang tror jeg nok de gjør ting mye bedre. Kanskje bare ikke den erfaringa var tilstede som de trengte. Jeg kunne selvfølgelig undersøkt: har det her firmaet den erfaringa for det valget, men i analysefasen var det ikke sikkert at man skulle bruke den og den modulen engang. Den tegninga jeg viste her, hadde tusen forskjellige andre forslag før det. Så det var litt vanskelig å se akkurat det her. En ting som kunne ha vært lurt her, hadde vært å sjekke hva øvrigheten mente om sikkerhet, på de her produkta de hadde. For det er sikkerhetshull i ASP-plattformen som ikke er anerkjent, blant annet i organisasjon A, så er ikke sikkerhetshull som det her anerkjent, du kan ikke bare innføre det uten videre i maskinparken. Det må spesielle hensyn til.

A: Synes du de fikk litt for frie tøyler når det gjelder...

S: Nei, vi hadde ikke noe bedre å stille opp med sjøl, i grunnen.

A: Nei vel.

S: Men i ettertid veit jeg vi kunne brukt spesielle avdelinger i organisasjon A som har særdeles god peiling på sånn. Vi prøvde, men hadde ikke nok forankring.

A: Men skulle du ønske at dere hadde gjort det på den måten i stedet for?

S: Vi skulle ha vært mer nøye, vi skulle ikke tatt det for god fisk helt i starten. Vi hadde et skikkelig møte på det her, hva er det man kaller en sånn gruppe, ikke ressursgruppe, men en sånn gruppe som i grunnen ikke har noen sånn ressursavtale om at de skal være med fra starten til slutt, men de er der som guruer akkurat der og da. Ut i fra sitt eget fag.

A: Konsulenter?

S: Ja, på en måte bruker vi de som en slags konsulent. Og flagge oppgaven til dem på en måte som sier at “dere skal inn, løse oppgaven, å gå ut”. Hadde vi fått forankra det her skikkelig godt, og de hadde forstått alle konsekvensene, så hadde vi kunne gjort noe med teknologivalget.

A: Okei. Ville forbedringer tror du kunne vært gjort for at prosjektet skulle ligget litt nærmere estimatene?

S: I grunnen så virka det som om det ikke var mulig å oppnå det her med de første estimatene uansett. Fordi at når man ser på resultatet, og ser på det nye estimatet, så er det så mange prosent mellom de, at i ettertid så ser det ut som en utopi. Det gikk ikke an å nå det. Når det her i tillegg var time for time, så var det ikke tatt så seriøst. Nå når det her var tatt på fastpris, var estimatene tatt jææækla seriøst. Det var samme person som gjorde det, personen er jæklig god, han er av deres beste på akkurat det feltet der, og det er bare at nå tok de seg sjøl i nakken og gjorde en skikkelig detaljspesifikasjon, og ut i fra der gjorde de estimatene, ikke andre vei. De var sikker på at de hadde alle ønskene våre på bordet, så gjorde de forsøk på detaljspesifikasjoner, så gjorde de estimat.

A: Mmm. Så dere har fått en mer sånn kontrollert....

S: Ja, i rett rekkefølge. Klare milepæler og sånn at vi i grunnen ikke skal gå tilbake når det gjelder å låne bort penger.

A: Okei.

E: Dette høres egentlig ut som om du mener at, å jobbe iterativt, det er jo noe med det at da har du ikke en veldig detaljert spesifikasjon som du baserer estimater på, spesifikasjonen din er på en måte en del av leveransen som du lager underveis. Og det var kanskje det som skjedde i den første fasen av TelMont, at krav og design og sånn ble jo iterativt utvikla, så selvfølgelig vil jo det være en situasjon hvor det opprinnelige estimatet vil være mer usikkert, men dersom man kommer med en detaljert spesifikasjon, og så (slår i bordet) stempler den på en måte, sier “det er det vi skal lage”, så lager et estimatet basert på det, er det ikke det som egentlig er litt forskjellen her? Jeg lurer på, vi har vel egentlig vært ganske mye inne på det nå, men altså, nå begynner jeg nesten å spørre om politikk her, men er det rimelig å si at ansvaret for det estimatet kun var på leverandøren, for å si det sånn, hvis man da skal jobbe iterativt og inkrementelt, sånn at kravene ikke var helt fastlagt på forhånd, synes du da det er rimelig at estimatene da kun er .... at det kun er leverandøren som har ansvar for estimatene?

S: I den prosjektformen vi hadde når vi starta i hvert fall, så var det IKKE rimelig at de hadde hundre prosent ansvar for estimatene, i den prosjektformen vi fikk etter hvert, så var det VELDIG rimelig at de hadde ansvaret. Beklageligvis. Hvis det her skal bli en nyttig erfaring for prosjekt, så bør man se på: “Har man en intern eller

en ekstern leverandør?” Butikken endres veldig når du skifter der, altså. Og har du en intern leverandør, så tror jeg fullt og fast på en sånn skikkelig iterativ prosess, og skal du som prosjekt prøve å tukle med pengene til sjefen din, bør du fortelle sjefen din at her kjører vi en iterativ prosess, og kan hende ender vi opp med mye mer enn det som var forespeila.

A: Mmm.

S: Sørg for å få et SKIKKELIG godt tak å jobbe under, sånn at du slipper å komme hylende tilbake hver gnag du har vært i møte med den andre. Skal du da ha ekstern leverandør, så er det i grunnen et forhold mellom leverandøren og sjefen din, og da bør leverandøren ha gitt et ganske godt estimat. Så her er to erfaringer som gjelder for to forskjellige organisasjonsstrukturer. Ta et vanlig EDB-firma som har en god ide, kan godt finne på å prosjektere og å lage hele greia sjøl. Først lage et produkt fordi de ser det er behov på markedet, og så kommer de ut med det, da har de idehaver og alt sammen innenfor de samme veggene.

A: Hva tror du er årsaken til at estimatene ble oversteget, sånn helt konkret?

S: Jeg lagde jo en tabell over det....Årsaken var jo mye på grunn av teknologivalg, og endringer her og der som ble “snikinnført”.

A: M-hmmm.

S: Skal vi se om vi finner noen tredjeplass her....Man satt ikke ned foten tidlig nok når man så at her var det overskridelser på gang. Man gjorde ikke noe med prosjektet, man lot det skure og gå. Man fikk vite gjennom fakturaer som kom nesten en måned for seint, at her har vi hatt en overskridelse. Så blir det holdt et krisemøte hos leverandør, der de ringer oss og forteller akkurat det samme som vi fikk faktura på for noen dager siden. Ikke helt bra måte å følge opp på, og de statusene vi fikk tilbake, de var hele tida tekniske statuser, sånn kan vi gjøre, sånne løsninger har vi, da og da, men utviklinga var de litt mer hemmelige om, i grunnen.

(Snur kassetten)

A: Hvordan synes du det var å ha tilgang til applikasjonen underveis gjennom hele prosjektet, sånn som...

S: Ja, de har presentert en tilgang til applikasjonen hele tida, altså, de brukerne som hadde tilgang til applikasjonen hele tida har hatt det kjempefint. Som prosjektleder var det så mye annet å ta seg til at hvis du begynte å tukle med det der, så var det en viss fare en plass, at du tråkket i salaten, noen andre sin salat, det var ikke ditt bord. Og de gangene jeg prøvde å se littegrann på det, rett og slett fordi jeg synes det var jæklig artig, jeg er jo ingeniør sjøl, og synes det her er artig, så gikk jeg en smule uforberedt inn i problemstillinga, enten så skapte jeg problemstillinger om ikke fantes, eller så skar jeg gjennom noe altfor tidlig eller noe sånt. Så hvis man skal være prosjektleder, så bør man kanskje bare gi slipp på akkurat den her 7. far i huset-rollen her, og bare nekte å ha tilgang til det. Så nå har jeg ikke tilgang til testsystemene en gang. Jeg vet ikke hav som finnes inne i dem, en gang, Og det vil jeg ikke ha.

A: Nei.

S: Og det er fordi vi kjører det her veldig rigid nå. Men hadde jeg vært mentor på heile analysen, alle detaljene og sånn, så hadde det vært artig å hatt tilgangen til det. Men da hadde det vært .....Da hadde jeg turt å gjort det fordi resten av prosjektet kjørte seg fram av seg sjøl. Men den gangen her står jeg bare som påhengsmotor, og ikke som ballastvare.

A: Hadde dere noen sånn helt klar avtale om hvordan produktet skulle se ut når dere startet utviklinga?

S: Du at meg jo littegrann der. I grunnen hadde vi jo ikke noen skikkelig avtale om at produktet skal se ut sånn når vi begynte analysen. Men vi var enige om hvilke komponenter som skulle stå på den tegninga jeg tegna her, og så hadde vi i grunnen ganske klare krav. For at hver av de to underleverandørene skal utvikle noe, så skal de jo ha beskjed om det et halvt år i forveien, omtrent. Det er ganske rigid der. Og dermed så hadde vi levert kravspekker der, vi fikk detaljspekker tilbake, i hvert fall fra de, på en sånn måte at vi så at: nå har vi lagt rammer, vi leverte i september, utviklingsprosjektet, starta i oktober. På den måten så hadde vi ganske klare forventninger om hva det her skulle bli. Første forventningen, for alt i verden, tilpasse oss til det vi kom til å få fra Y og Z (underleverandører), for vi får ikke gjort noen endringer der på null tid. Og så var det da grensesnittet der, da. Kanskje ikke vi hadde klare nok forventninger om hva grensesnittet skulle bli.

A: Okei. Mmm. Er det noe du vil anbefale å ha?

S: Vi har snakke om det her med en iterativ prosess fullstendig lukka inne i et firma, eller om det er mellom to firma, jeg ville kanskje ikke anbefalt å ha en sånn klar avtale om hvem man skal ha hvis man...at man er inne i et firma. Da lager du en kravspekk, og hvordan du i grunnen utarbeider kravspekken, da kan du godt legge inn det utviklingsfirmaet til å hjelpe deg med å lage en kravspekk, kall det en analyse, ta det gjerne time for time, men når vi er ferdig med det (plystrer) (feier med hånden) levert kravspekken, da har vi et utviklingsprosjekt på gang. Ingen veit hvor haren hopper før du er ferdig med kravene. Og kravene våres, de var spikra kanskje littegrann for seint. i første trinnet. I andre trinnet så var de jo ferdigspikra. Men nå skal jeg si en ting som ikke har vært sagt ennå, at gjennom trinn 1, så var det veldig mange tanker som var lagt til grunn for trinn 2. Mage tanker som var helt sånn, ferdig med det, få det inn i kravspekken, få det ut. De gjorde ferdig de kravene utrolig fort, samtidig som de endringene vi var nødt til å ha i neste fase hos Y var gjort av dagens prosjektleder hos organisasjon B på null tid. Han var meget god person, dyktig person, og gjorde det her utrolig raskt. (Uklart)

A: Synes du prototypen de lagde var med på å finne ut hvilke krav dere trengte?

S: Ja! Det synes jeg. Prototypen var nyttig - absolutt. Og skulle man kanskje ha passa på prototypen veldig bra, så skulle man kanskje hatt et lite... en liten milepæl her, men jeg er ikke helt overbevist om hvordan jeg skulle formulert den milepælen, i grunnen. Har dere noen forslag?

E: (Ler - og trekker på det.)

A: Men jeg fikk inntrykk av at de synes den prototypen var for stor, men synes du den var for stor? Den første prototypen som kom i januar? Som skulle evaluere teknologien og sånn.

S: Nei, den var...jeg synes ikke det. For den la, altså, da la den grunnlaget for hva som skulle utvikles. Jeg synes at det gjorde ingenting at den var så stor. Og når vi snakka om prototype, så snakke vi jo om å få den ut i januar, og jeg synes ikke det var noe rart i det heile og det store.

A: Nei vel.

S: og da hadde du januar på deg til å evaluere prototypen, levere inn endringer, og få noen folk til utviklinga skulle være ferdig. Og klar til test, ikke sant.

A: Mmm.

S: Og da viste det seg jo at det var masse endringer på prototypen som da kom inn. De var snikinnført. Kanskje man burde hatt en mye mindre prototype, men da mye tidligere. I grunnen. Fordi at når man lager estimat, så lager man estimat på en detaljspekk, en utvikler vil ikke tenke detaljspekk uten å utvikle, skal han

utvikle, så lager han prototype. Bare for å tenke bakover.

A: Mmm.

S: og da, den prototypen der, vi kunne ha presentert den og hatt en designreview på den, det hadde vært veldig fint, bare vi hadde fått den opp litt tidligere.

A: Synes du at prototypen som kom i januar hadde for mye funksjonalitet, eller var det passe?

S: For den saks skyld var den kanskje veldig detaljert for å...bare for å verifisere at de boksene her fungerte i hop, så var den kanskje veldig detaljert. Men vi hadde et problem. Prototypen kom før noen av de her i det hele tatt hadde en leveranse. Så vi hadde jo ingen peiling på om de funka i hop. Prototypen fungerte da ikke i lag med medhjelperapparatene, det fantes ikke noen medhjelpere. Og der har vi i grunnen en utfordring, som var ganske vanskelig. Altså, prototypen mista poeng-et sitt, det er nesten så du kan si det, for prototypen skal være et slag "proof of concept", ikke sant, og når du ikke har noen konsept å prøve den på, så har du kanskje litt vanskeligheter for det. Det som er igjen av en prototype da, det er jo nesten bare grensesnittet igjen. Og når de presenterte grensesnittet, så måtte de nesten presentere det med alle de detaljene det har, i grunnen....

E: De fikk jo testa ut en del COM og ASP og...

S: Ja, plattformene fungerte, og det er klart. Det er helt riktig, det er kanskje det som er hovedformålet til prototypen egentlig og, å få plattformen til å fungere....

E: Vi kan jo ta inn litt fra den analysen vi har gjort. Blant annet så mener jo vi da, ut i fra de intervjuene vi har gjort, at prototypen ble litt for stor. Og det var også p å en måte en sammenblanding mellom en kravprototype og en teknologiprototype.

S: Ja, det var klokt. Det var klokt. Ja, for det hørtes jo faktisk sant ut, for, ja, som jeg sa i stad: "da er det bare grensesnittet igjen" - den er jo kanskje ikke riktig den setningen jeg sa. Den setningen jeg sa. Det som står igjen, det er plattform, og grensesnittet. Og grensesnittet har med krav å gjøre, for at vi setter ingen krav om plattform, og så har du plattform. Prototypen, kanskje målet med prototypen burde ha vært å få de store boksene på tegninga til å fungere, og få plattformene til å fungere, og så kanskje da....man burde ha skilt prototypen. Det hørtes ikke så dumt ut....

E: Ja, for at det vi har på en måte foreslått, er jo litt det at det ble så mange krav i den teknologiprototypen, på en måte, at innen vi var ferdig med å evaluere teknologien, så var prototypen blitt så stor at man nesten ikke kunne kaste den. Man kunne ikke endre valg av teknologi, for det hadde blitt så mye funksjonalitet, slik at.....Er du enig i det?

S: Ja, det er jeg helt enig i. Det er jeg helt enig i. Veldig bra. Nå lærte jeg noe. Det er i grunnen å få satt ting.....begreper her, altså. En prototype er ikke bare et halvferdig produkt. Men tror du at hvis du hadde hatt mange små prototyper som hadde levert forskjellige ting, at du da ville fått et bedre (uklart)

S: For da ville vi kanskje kunne fokusert på plattformen. (uklart) jeg synes det aller viktigste jeg lærte nå, det var at da har du sjansen til å gå tilbake. Du har sjansen til å finne at her funker det ikke. Og gå tilbake. Du har ikke balla på alt mulig som du i grunnen kan prioritere etterpå. Det lærte jeg. Veldig bra. (Ler).

E: Målet vårt er jo å lage en erfaringsrapport som også de som deltok har nytte av.

S: Mmm. Jaja, kjempemessig.

A: Over på kommunikasjonen mellom dere og organisasjon B: hvordan var den?

S: Kommunikasjonen mellom oss og...Ja.....Altså.....Jeg hadde nå et møte med noen av de herre mentorene, de rareste personlighetene i hele organisasjon A samla på



ett bord, og de protesterte jo ikke engang. (Ler) Det har vært vanskelige sånne person.....Vi har en sånn.....Han er ikke utdatert på noe vis, han har utrolig mye futt og energi i den karen her. Det er han som har vært med i store prosjekter før. Før så var han.....Mister Automatisering. Det var han som automatiserte alle de sentralene som foreldra deres har prata i..... Men han er over 60 år, og har de.....ikke sant, han er veldig sånn (knipses) klar og grei, sånn "dette her er mitt ansvar", og da tar han ansvaret, pluss at den her karen er.....han kan bli.....han kan være vanskelig og forholde seg til, rett og slett, men.....

E: Var det han som var tidligere montør, eller....

S: Ja, ja.

E: At han var litt sånn initiativtaker på det prosjektet her?

S: Nei! Han.....han er det ikke snakk om. Det er en annen person som...som var med på møtene og fikk vridd fokuset hele tida. Så om vi hadde fokus eller ikke, så vrei han alltid fokuset til noe annet hele tida. Han tok inn mer detaljer enn det var nødvendig. Det kan dere skrive i rapporten, at her var det.....her kom detaljer inn som ikke var på tide å ta opp. Det kan dere skrive i rapporten. Og så hadde vi han tidligere montøren, da, han var en...han var en sånn.....

E: visjonær.

S: Det var godt ord. Veldig bra. Han var en sånn visjonær. Altså, det har jeg lest en plass, at du skal ha i prosjektgruppa all verdens personligheter, du skal ha en sirkusartist, du skal ha en positiv og du skal ha en negativ, og det hadde vi i høyeste grad i den her gruppa her. Og det var kanskje veldig sterke. Jeg var nok mere positiv, sånn helt uansett, skjønner ikke engasjementet hvis det er noen som synes noe er trassig, det får vi ta når den tid kommer...mens andre gjerne bare vil snakke om problemer. Noen er problembasert, andre er løsningsbaserte. Og så....samarbeidet mot organisasjon B...eh...det va en litt spesiell protokoll og...noen hadde sånne spesielle protokoller. Jeg kan nok bli vant til det, men...og...og...jeg var vel og kanskje en som gjorde prioriteringer.....sånn, sånn, sånn gjør vi. Det var kanskje ikke til stede som en egenskap der.

A: Nei vel. I hvor stor grad følte du at du hadde kontakt med utviklingsteamet mens de jobbet med.....

S: I grunnen for lite. Utviklingsteamet savna mer personlige telefoner fra meg. Det var kanskje fordi de ville jeg skulle bry meg mer om hva de jobba med.

A: Mmm.

S: Men.....

E: Mener du da utviklerne, eller mener du....

S: utviklerne ja, ikke prosjektleder. Prosjektlederen hadde jeg en god kontakt med. Vi hadde en.....altså, du har mange måter å kalle en kontakt for god. Vi hadde.....vi kom godt overens, vi kunne faktisk føre en krangel uten også slenge på røret, og sånn, men likevel var det mange ting som jeg sa tidligere som ikke kom fram, og da var den oppfølginga som.....og hvis du skal prøve å ikke la det formørke forholdet oss to imellom, så hadde vi i grunnen et forhold som begge mestra ganske godt. I grunnen.

A: Mmm. Eh....over til det neste. Følte du at dere fikk et riktig produkt? Du har jo sagt det, men hvor fornøyd var dere med det, sånn i forhold til.....

S: Produktet, det som vi har på pulten her, det er det trampeklapp for. Folk er veldig fornøyd.

A: Men var det sånn som du hadde tenkt deg det, eller er det helt...

S: Det var nesten som jeg hadde tenkt meg det.

A: Mmm.

S: Men hvis vi nå skal snakke om at det er en webapplikasjon, så kom jo den ideen etter en liten stund. Fordi at.....at det var noen sjefer som trodde at det her, det kan en løse på Excel. Og hente inn eksisterende (uklart) fra støttesystemet. Støttesystemene hadde ikke de rapportene, dermed ble det ikke (uklart). Du kunne ikke kverne noen resultater, nei, noen data du ikke hadde. Å bruke data på denne måten her forutsatte at du brukte visse typer grensesnitt, jeg kunne se det som en mellomvare, det var ikke mulig å kjøre her. Derfor så balla det her på seg. Du henter noe ut i fra de store, tunge, rigide stormaskinplattformene der, så har du nødt til å ha noe tøffere enn bare en sånn her excelgreie. Og særlig fordi du hadde ikke de rapportene der. Tidligere prototyper på TelMont, det var ikke oppfatta som en prototype, da, men det var en excelting som utnyttet eksisterende rapporter. Og den fungerer bra. Så den modulen som utgjorde.....eller den versjonen vi utvikler nå, så har vi faktisk en egen excelmodul som er utvikla hos oss sjøl.

A: åja.

S: den henter eksisterende rapporter fra systemet, og som leverer ferdig kverna materiale til TelMont. Så den har vi kjørt her. Og den er og kjørt veldig bra.

E: Men jeg tenkte på et sånt oppfølgingsspørsmål her: akkurat det med det på en måte.....du sier trampeklapp for applikasjonen, og det er på en måte et inntrykk vi har fått også fra.....at på en måte kunden, det vil si dere, er fornøyd, og brukerne er fornøyd med produktet.

S: Ja.

E: men jeg lurer på litt....han visjonæren, på en måte, han tidligere montøren, er han også fornøyd? Jeg har fått inntrykk av at han ønsket seg et ganske my større og mer fleksibelt system enn det egentlig sluttbrukerne trengte eller ønsket seg.

S: Han har brent seg på IT-systemer før. På en spesiell ting. At dukker det opp en endring i organisasjon A som er (uklart), så får du ikke leverandøren av det her produktet til å gjøre (knips) en sånn her endring. Du at noe på en tekstfil, endrer den tekstfila, slik at du rett og slett tukler med databasen som en bruker greier å gjøre uten å kunne gjøre noen skade, så er det veldig bra. Og det vil han gjennomføre, og har fått gjennomført det til en viss grad, så nå når vi kom opp med den egne excelmodulen vår, så var han veldig grei. Da så han at prinsippet hans var verdifullt. Men det her er en visjonær, og jeg var klar over at han visjonæren var en veldig sterk personlighet, men jeg tror han og er fornøyd med produktet. Han kunne ikke.....han innså med en gang at hans ideer kosta penger, og da var det intern øksing om..om.....om vi skulle ta med dem fordi det kosta penger. Det hørtes naturlig ut at det kosta penger, ikke sant. Det at han har uttalt seg fritt og sterkt mot organisasjon Bs prosjektleder, i den grad de ikke har takla det, det er fordi de ikke takler.....at da har de ikke takla tøffe personer. For at det er bare å takle tøffe personer, så .....kan du forstå store ord og bruke store ord, så bør de ikke føle seg støtt av han, eller oppfatte han som særdeles misfornøyd. Han bør ikke være misfornøyd, for det var han som tegna grunnflytskjemaet på det her. Og flytskjemaet er på en måte gjennomført. De grunnideene hans er....

E: Ja, jeg sier ikke at han ikke er fornøyd med....

S: Nei, nei. Akkurat.

E: Det er min, dette her er mine tolkninger.

S: åja, akkurat.

E: Det var ikke direkte fra organisasjon B det her.

S: Neivel. Ja, OK. Men sånn sett så er han som enhver annen femtiåring med en

del livets erfaring, og en del faktisk gode erfaringer og. Og.....og at han er en sterk personlighet, det er helt klart. Man kan ikke komme fra det. Jeg har et inntrykk av at han er fornøyd med den posisjonen han har nå, det vil si, ganske på sidelinja. Og han brukes kun som en visjonær, så jeg forstår at han...Og jeg tror ikke han er noe sår for at alle....at noen av ideene han bare blir kasta bort, fordi det kosta penger, ikke sant. Eh....

E: Det er jo blitt et ganske mye enklere system enn dere.....et mer strømlinjeforma system enn det han ønsket seg...

S: Ja,ja. Det er helt sant. Men.....men.....hvis man vet at man har en visjonær rundt seg, så må man jo bare innse at det files og gnikkes litt på. En sånn en er nødvendig i en fase.

A: Mmm. Men han var veldig mye med på, sånn som jeg forstod, den initielle kravspesifikasjonen, hvor fornøyd var du med den?

S: Den initielle kravspesifikasjonen var jeg jo ganske.....jeg var ganske, jeg var helt fornøyd med prinsippene. Men alle verdens.....jeg er ikke fornøyd med at vi ikke hadde tenkt på alle de detaljene det faktisk går an å tenke på. Men det var fordi jeg ikke visste at det gikk an å tenke på de. Så skulle jeg ha skrivd det her på nytt igjen, så hadde vi selvfølgelig ikke vært fornøyd med å slippe fra med....det er som å slippe en tolvåring ut i det voksne livet. Ikke sant.

A: Mmm.

S: Men vi visste ikke at det mangla noe fra 12 til 18. var nesten ikke klar over at man skulle i militæret først engang. (latter). Så....

A: Så de endringene som kom, de var nødvendige de da kanskje?

S: Ja, de var nødvendige. (uklart, pga alle pratet i munnen på hverandre.) og det kutta av en del her og der var og nødvendig....

A: Ja, det var også nødvendig. Jeg forstod at den var ganske brei, men så kom brukerne inn som skjærte ned på det ganske mye.....

S: Ja.

A: Og det var også nødvendig?

S: Ja, det var også nødvendig. Så den prosessen vi til syvende og sist oppnådde, med at visjonærens ideer var nøye detaljert.....nei, nøye fundert over av brukere, der vi tok rede på restene av kollisjon, og bygde rundt det, det var i grunnen veldig bra. Og da hadde vi og et problem, ser du, brukerne, som vi var veldig avhengige av, fikk vi slettes ikke lov til å bruke, ledelsen deres trakk til bake brukerne. De ville ikke....de ville ikke se at de...at de...at de "drev på med sånn her tull". De skulle jo jobbe og slite med de gamle systemene, for det tok alt for lang tid å drive på med det de gjorde.

A: Ledelsen i...?

S: Deres personlige leder.

A: åjaja.

S: At vi representerer eierne av telefonsentralene, de representerer de som drifter det her, det er ikke de som er teknikere som sitter med mann-maskin- kommandoer og sånn. Men det er de som kjører blant annet... de er superbrukerne.

A: Mmm

S: Og de folka her var nærmest, nesten dratt inn tilbake til han, altså, fordi at det var.....de likte ikke at det her var....at de var putta til det. Men det her er snakk om en organisasjon (uklart). For lederne forstod ikke at de var nødt til å være med i det prosjektet. Lederne hadde skrevet under på at de skulle oppfylle...oppfylle de kravene som var satt i beslutningsdokumentet. Det vil si i businesscasen. Men de

fikk seg likevel ikke til å hjelpe oss å oppnå målene.

A: Så først så var det problemer med å få tak i brukerne, og så var det problemer med å holde på de?

S: Ja! Ressurseierne var vanskelige, men det virka som i etterkant at ressurseierne er ikke noe problem nå når de ser at, at vi faktisk er nødt til å bruke der her...

A: Så det er vanskelig å få kunden, nei, eller brukerne til å forstå at de er viktige, da?

S: Ressurseierne, det vil si sjefene til kunden, vil ikke forstå det her.

A: Nei.

(Prat i munnen på hverandre)

S: Jo, i høyeste grad, det har vært en krangel i mange år om hvem som skal være prosjekteiere i det her.

E: De ønsket det altså?

S: De ønsket at noen andre enn de sjøl var prosjektledere, at alle ville jo at de skulle være prosjektledere for at det var de som eide de brukerne, ikke sant, men.....men.....den her gangen så var det jo noen andre som sa: okei, da tar vi det. Men det her blei justert internt i organisasjonen etter hvert, vi justerte på den organisasjonen, og fikk tjora det her til.

E: Men akkurat dette her synes jeg vi må få klargjort litt, for dette her, nå synes jeg vi er på et kjernepoeng her.

S: Ja, veldig kjernepoeng. Jeg er ikke sikker på hvor mye jeg har lov til å fortelle, men inntil videre tror jeg at det er greit.

E: Men igjen så sender vi deg....

S: Jaja, bra.

E: Men, altså, eh...initia.....det er to grupper etter som jeg har forstått. Du, som på en måte representerer de som skulle.....

S: Ja, jeg representerer prosjekteierne, og ressurseierne.....

E: Som sagt, det jeg er litt interessert i å vite, var på en måte hvem det er som var initiativtakeren? Altså, jeg synes det er så rart at hvis de som satt med de brukerne, altså de som var ansvarlige for....sjefene til brukerne...hvis det var de som ønsket systemet, så synes jeg det er rart at de ikke innså at de også måtte tilgjengeliggjøre ressurser, det vil si, de sluttbrukerne.

S: Mmm. Vi har en litt vanskelig historie med hvordan det her prosjektet her var starta opp. eh.....jeg tror ikke det her er representabelt for organisasjon A generelt. Derfor går det kanskje an å fortelle om det.

A: Mmm.

S: Eh.....Jeg har forklart at det er vi som gir, altså mellomsjefene gir de sjefene et oppdrag som deres ansatte skal utføre. Og krangla om hvem som skal levere de ansatte et bedre verktøy. Så fikk de da endelig starta opp det her, da, og man sier, flott, få til et verktøy, få det til å skje fort, gjerne jækla fort. Så i grunnen så var størrelsen på det her prosjektet, det var ingen som ante om størrelsen på det her prosjektet når man satte ned en gruppe. Og jeg forstod i grunnen i utgangspunktet ikke hvor mye man hadde. Hvor mye man hadde tenkt i dimensjoner rundt det her i utgangspunktet, så jeg regner med at det kom som en overraskelse på de aller fleste sjefene at det her var så stort som det var. og....og her er det snakk om å beskytte sin egen rygg. Hvis jeg har bestemt at jeg beslutter at jeg skal kjøre et prosjekt som er så og så stort, så må jeg ha så og så mye penger, og jeg må ha så og så mye gevinster, skal jeg få gevinster, er jeg nødt til å få noen til å love å utføre de gevinstene. For det var ikke jeg som skulle sette meg ned som bruker etterpå og bruke

verktøyet. Så her er det snakk om å beskytte sin egen rygg. Det vil si å få de sjefene til å love at de brukerne i grunnen skal utnytte verktøyet på en god måte. Det var ett av skrittene som gjorde at de og var nødt til å stille mer ressurser. Det var ikke helt klart i utgangspunktet. Det hadde ikke organisasjonen som helhet tenkt på, i grunnen. Og jeg tror det her var litt spesielt, altså. For organisasjon A. For det er ikke mange ganger det er starta opp et nytt støttesystem. Det skjer veldig sjeldent, i grunnen. Så her var det en unik sjanse. Som regel så, altså, det organisasjonen våres i hvert fall er veldig god på, det er å kjøpe disse enorme telefonsentralene. Det er noen helt enorme system. Det er de veldig gode til. Og til det har de helt tilpassede prosjekter, og...folk som har et enormt kontaktnettverk. Enorm erfaring. Men overraskelsen var jo da til syvende og sist, da jeg hadde tatt rede på, vi fikk visjonæren til å ta kontakt med brukerne, vi tok rede på restene av kollisjonen, og bygde videre på dem, vi engasjerte kanskje organisasjon B altfor tidlig i forhold til å få ferdig oppryddinga etter den her kollisjonen. Det er kanskje noe...det er noe jeg har sittet med inntrykk av. At vi burde engasjert dem på et mye seinere stadium.

A: OK. Mmm.

S: Men organisasjon B kom inn ganske tidlig og begynte også hjelpe oss å tenke IT. For det er organisasjon B som kan tenke IT, og som fortalte at her er det en del forutsetninger som må legges. Og der gjorde de en god jobb. I den grad at en uerfaren IT-utvikler som hjalp til med mye i startfasen her. (uklart) og de fortalte littegrann om hvordan det her skal drive....altså, hvordan ser systemet ut, ga en visjon på hvordan systemet skulle se ut når det stod ferdig. Men den erfaringa de kom og fortalte oss, den veit vi nå vi kunne henta fra mange andre plasser innenfor organisasjon A, og da kunne vi unngått å få den her rotete utviklinga helt i starten av prosjektet.

A: Men ville dere heller hatt organisasjon B samtidig som dere hadde brukerne der? Litt seinere i prosjektet?

S: Ja, altså, jeg ville hatt brukerne først, og så organisasjon B. Det var feil rekkefølge, rett og slett. Ja!

A: Det er noe generelt du kan anbefale?

S: Ja! Det kan jeg virkelig anbefale. Altså, få gjort ferdig alle visjonære ideer, og få gjort ferdig alle interne kollisjoner før du går ut og spør en utvikler. Og det er jo helt klart også....det ser man jo overalt. Norge kommer som en organisasjon som skal hjelpe et land i borgerkrig, og så kommer vi inn og prøver å hjelpe dem, men forholdene skifter seg hele tiden under borgerkrigen, vi har ikke sjanse til å hjelpe dem så godt som vi tror. Derfor må vi bare sende milliarder på milliarder nedover til land som ikke greier å ta rede på de tinga der, ikke sant, på mange måter så burde vi kanskje ha rydda opp hos oss sjøl før vi betaler 1000 kroner timen til en konsulent.

A: Ja. Mmm. Synes du brukerne var for lenge med i prosjektutviklingen, sånn at det blei for mange endringer i krav?

S: de hadde ikke.....de hadde ikke....uff, vi ga ikke de noen helt soleklare mål på når de skulle være ferdig, det var kanskje en tabbe av prosjektledelsen, rett og slett. Men det var jo mye fordi at det var jo ingen brukere å få snakke med. Brukerne vi fikk snakka med, var jo jækla midlertidig. og....ja.

A: Men det ble jo veldig sånn, slik jeg har forstått det, så hadde dere en kravspesifikasjon, og så kom brukerne og så ble det masse endringer, så det kunne jo syntes organisasjon B at brukerne var for lenge med, da. Men syntes du også at det var et problem, at dere holdt på for lenge med kravspesifikasjonen, eller måtte det bare

bli sånn?

S: Det måtte bare bli sånn, littegrann fordi ingen av oss hadde sto peiling på IT-utviklingsprosjekt på forhånd. Hadde vi kunnet den teorien som dere driver og studerer her, så hadde det gått mye bedre. Hadde vi hatt rapporten din nå, så hadde vi fått det her mye bedre til.

A: Ok.

S: Men da så hadde vi ikke de rette ressursene. Rett og slett, og de ressursene som var inne var ikke i stand til å se at vi måtte ha flere ressurser. Det var en ressurs som på en måte kunne deltatt littegrann, men det var en del sånne forutinntatte meninger her som det ikke gikk an å forene, så det samarbeidet, det slutta lenge før det var fruktbart. Dessverre.

A: Mmm. Men underveis under prosjektutviklinga, følte du at du mista kontrollen over kravspesifikasjonen, eller hadde du.....?

S: Nei, jeg hadde ikke noe hundre prosent kontroll, nei. Det hadde ikke prosjektleder hos organisasjon B heller.

A: Nei. Hvordan synes du den dokumentasjonen og endringene av kravspesifikasjonen fungerte, synes du?

S: Den var dårlig.

A: Den var dårlig.

S: Endringer var ikke flagga på forhånd av mellomledelsen, og ingen hos oss var i stand til å fortelle hva det her kosta. Estimater var rett og slett ikke helt tatt rede på. Man tenkte ikke estimat når man tenkte endring.

A: Kom dere med noen forslag til endringer, da, underveis i prosjektet?

S: Ja!

A: Det gjorde dere.

S: Det var ting som ble sett på som rart og vanskelig, og nå snakker jeg...nå kommer vi tilbake til det her med at skjermbildet, altså prototypen, var ikke en spesifisering, men kun et forslag. Det betyr at brukerne våre hadde rett til å plukke på og angripe skjermbildene så mye de ville. De så at det var mye som var tungvint, tre skjermbilder for å få gjort en ting, så det ble gjort. På den prototypen som vi hadde på og utvikla....

A: Så det fungerte bra?

S: Ja! Og det var vi først i stand til å gjøre når vi hadde kommet fram til den litt for store prototypen, som var i januar. Kanskje...altså, den kongstanken der om å dele ting littegrann på prototypen, hadde kanskje hjulpet littegrann her.

A: Ja...Hvordan vil du gå fram når du lager kravspesifisering i neste prosjekt?

S: Eh.....det jeg sa i stad var i grunnen å ta rede på kollisjonen mellom visjonær og bruker, og få ei kjapp vurdering om hvordan det her vil se ut i et IT-system, kanskje og få en budsjettpris. Det vil si en uforpliktende pris som leverandør gir for at vi skal ha noe å føre på budsjettet.

A: Ja, mm. Var det noen roller du synes var overflødige i prosjektet?

S: (Latter) Ehm....Det er mange som har hatt flere roller samtidig, så det er litt vanskelig å svare på. Prosjektet har ikke vært så stort at vi har hatt helt soleklare roller, men.....men det var littegrann med det her at du skal vite hvem sitt klaver du tramper i. Det er kanskje lurt, i hvert fall hvis du samarbeider med folk som er litt eldre enn oss, kanskje dobbelt så gamle. Da skal du være ganske klar over hva du driver på med som menneske. Du ser jo, altså, det handla kanskje litt om livets skole, det der, ellers.....ellers....når du har en klar rolle så har du jo en, så kan du jo sørge for at du har den rette motivasjonen. Men jeg ser...jeg ser jo at

når ressurseierne trakk tilbake sine ressurser, så mista de jo motivasjonen. Men den rollen....men det var noen roller som var overflødige i forhold til det.....de var ikke trukket tilbake fordi de var overflødige. Alle sammen måtte være der. De rollene som jeg kan huske at vi har hatt, måtte vi ha.

A: Mmm, ja, OK. Var det noen roller du savna, da?

S: Ja, i etterkant så savna jeg en som hadde særdeles god peil på IT-utvikling.

A: Ja vel.

S: og som visste om alle irrganger i organisasjon A. Og som visste om alle de...alle de mulighetene som finnes innen organisasjon A. Nå veit jeg de. Nå spør jeg folk mer om de mulighetene. Før så burde jeg ha spurt noen andre om hvem muligheter som finnes. Da hadde vi blitt litt bedre på IT-utvikling, tror jeg. I hvert fall hadde jeg og mine folk vært litt bedre da.

A: Men er det noen delaktiviteter du synes dere brukte for lite tid på?

S: Eh....på å spikre kravene, og vurdere konsekvensene av kravene. Brukte vi for lite tid på. Vi brukte for lang tid på å virre rundt før vi fant ut hva slags konsept vi skal lage. Konseptet kommer mange ganger før du får kravspekken, ikke sant, vi brukte litt for lang tid på konsept, litt for lite tid på å få en ferdig kravspesifikasjon.

A: Ja vel. Mmm. Jeg har ikke noe mer.

E: Nei, det er kjempebra.

S: Det synes jeg og. (Latter)

E: Er det noe du har lyst til å.....er det noe annet, altså, du veit jo omtrent hva forholdet vårt her er, vi vil samle så mange erfaringer som vi overhodet kan, og så prøve å lage praktiske retningslinjer innkorporert i Genova webprosessen.

S: Jeg har hørt ekstremt lite om Genova webprosess, sjøl om jeg har gjort en del anstrengelser for å vite om Genova webprosess.eh...så, jeg veit ikke om dere har anledning til å gi et lite lynkurs her nå, så får jeg si om det her prosjektet har misa fullstendig på den her modellen.

(Erik klargjør - kassett av)

S: Kanskje intervjuet mangler littegranne her, da. eh...vi har kanskje skrapa litt for lite, lagt litt for lite vekt på akkurat det her nå, men bare for å oppsummere: vi har sagt det at webutviklingssystem, bare man får kjørt en prototype der og får presentert det som skal på web, så har brukerne anledning til å sitte hvor som helst hen og samarbeide med utvikler, og bruker har når som helst anledning til å komme med en liten endring, liten endring som er veldig vanskelig å flagge oppover. Og en bruker veit ikke om kostnaden av endringen, men det er veldig mye enklere å komme med en liten endring når brukeren får det her presentert på web. For da trenger man ikke sitte i fanget til utvikleren. Og webgrensesnittet er mange ganger så begrensa at det er mange ting som er så såre enkelt at du kan forlange at en intelligent forstår nesten absolutt alt som foregår på skjermbildet der. Det kommer ikke noen sånne kryptiske tegn skrevet i Courier, grønn skrift på svart bakgrunn, men en sånn underscore der su skal fylle inn ting, sånn er de her gamle systemene som jeg tegna der for eksempel.

A: Mmm.

S: Eh....Eh...Og...Eh....Så fallgrubene her er jo i grunnen at vi kan innføre masse endringer nesten i.....en orgie av endringer, eh...fordelene er at du får det fort ut, en prototype må du nesten lage for å kunne presentere endringene dine. Prototypen blir til mens du går. Og her er kanskje litt utfordring med også skille mellom plattformprototype, brukergrensesnittprototype. Skjermbilder er ikke en spesifisering, men ett.....bare et forslag, det betyr at....at skal du jobbe med den biten der, så må

du allerede ha fått plattformprototype opp å gå, få den til å fungere skikkelig, før du kan presentere den med skjermbildene. Som prosjektleder, så bør prosjektleder kanskje ha tilgang til skjermbildene og, eh, fordi at han da ser hva som foregår, men det er vanskelig, eller det er kanskje skummelt å bry seg om detaljer. (Uklart)

E: Nei, men jeg tror altså vi har dekket det meste jeg.

A: Det kan jo hende du har noen erfaringer på hva som er forskjellen mellom det å utvikle tradisjonell applikasjon og webapplikasjon? Hva som klart skiller seg, lik-som?

S: Eh...ja. Et godt utviklet konvensjonelt system, har jo like enkelt for å endre ting. Men...Kanskje det...Men kanskje det er vanskeligere å se sammenhengen mellom businesslogikken og skjermbildene når det er snakk om en webapplikasjon.

A: Mmm.

S: Så har vi en helt annen diskusjon, og det er helt sikkerhet, eller tilgang, skal du presentere ting på Internett, eller intranett, det er to helt forskjellige ting. Skal du...altså, skal du presentere ting på web, så kan det hende at det blir på Internett til syvende og sist, selv om du ikke ønsker det i utgangspunktet. Da må du ta hensyn til de plattformene som butikken din i utgangspunktet har anerkjent og godkjent, det gjorde ikke vi. Eh....du må tenke mye mer på brannvegger når du driver på med web. Og hvis du skal utsette ting på en litt fancy måte. Det har vi i utgangspunktet ikke gjort, for den analyserapporten som jeg fortalte om som jeg har i sekken her, så vi hadde en kjempekollisjon med organisasjon B, fordi at de har ikke, de har ikke sett på alle behovene som skal til bare for å utsette bare en ekstremt liten ting på Internett, så det å gå ut på ået skikkelig Internett, det medfører kjempemye angående plattformer og....koden vår passer ikke, han må kanskje skrive om skal vi ta heile greia og så flytte over på Internett.

A: Mmm. Ok.

E: Det er vel egentlig sånn skaleringsproblematikk, man kan tenke seg noe sånt, da. Som er litt spesielt kritisk ved webapplikasjoner, altså overgangen fra intranett til Intranett.....

S: Ja, men vi snakker her om både sikkerhet og plutselig så skal du få godkjent det vi har tenkt å presentere. Du må tenke litt mer da på hvordan....på hva som er skummelt. For at det er mange ganger at du har lyst på noe fordi det gjerne blir enklere, du tenker ikke på at det er skummelt samtidig. Men at det er skummelt, det er helt soleklart. Og jo enklere ting er, jo skumlere blir det.

A: Mmm.

S: Så du må tenke littegrann der, hvem er det som blir begrensa sånn og sånn, skal du logge deg på, hvordan kan du begrense størst mulig. Men hvis at vi skal få noen til å logge seg på ået...en TelMont-ting, en agent som TelMont bruker av noe slag, at jeg skal ha passordkalkulator, det er et minstekrav, brukeren skal vite hvor han skal logge seg på, han skal kanskje være godkjent av de og de for å få lov til å arbeide med akkurat den og den arbeidsoppgaven, og så skal han kanskje vite enda et passord, så det er veldig mange sluser han må få til for å få lov til å logge seg på. Så skal han da i tillegg få lov til å utføre en enkelt ting. Og det har skapt kjempeproblemer i organisasjonen våres, fordi at de store herrer, de tror at det her er jævlig farlig for oss, det her er jo Internett. Det er en overbeviselsesprosess som jeg mottar hundre prosent uten organisasjon B, fordi at organisasjon B ikke var i stand til å hjelpe meg med det. De gjorde forutsetninger om at sånn og sånn kan du gjøre det, og så finner vi ut at: nei, det går ikke, men vi kan by dere en analyse på det her, men vi tar 300 000 for analyse. Men nå har jeg lært litt mer, og kan gjøre



det sjøl. Og bruker mye korter tid på det her. Så her har vi en terskel å passere når vi skal få ting ut på Internett. eh...vi må kanskje lage ting i ei, man må dele serverne inn i segmenter sånn og sånn, kanskje. For å passe på sikkerheta, og det er ikke all verdens programvare som fungerer på de og de (uklart). Det er sånn som i utgangspunktet var helt ukjent for oss, men som vi måtte ta mye mer hensyn til. Og det er ikke bare teknisk ute og går lenger, det er snakk om eierne og deres følelser for det vesenet her. Og igjen: brukerne ser at det her er veldig enkelt, det her vil vi ha, og de har ikke peiling på hva som ligger bakom.

A: Nei.

S: Så det må kanskje evalueres littegranne det her. Til å begynne med, så hadde vi lyst på WAP, vi hadde lyst på Internett, vi hadde lyst på at du kan gjøre ting fra PDA, og så bare det at PDA presenterer noe annet enn den fine oppløsinga på åen vanlig PC-skjerm, kan faktisk skape problemer. EH...WAP medfører jo mye uansett, man burde i hvert fall tenke på: skal man bruke SSL. Her er det mange forutsetninger som man kanskje ikke er klar over ennå engang. Så kanskje rapporten her burde finne ut hva slags forutsetninger som skal til for å disponere noe på Internett. Fordi at det er så skummelt. Det er ikke det at det her er noe terrorvåpen det produktet, men man skalerer deretter. Som om det skulle være et terrorvåpen.

A: Mmm. Ja, nettopp.

S: Det er kanskje litt spesielt. Så har vi det her med Genovaprosessen. Sånn som jeg trodde at Genovaprosessen var, så var det at man på åen måte skulle kunne skrive spesifikasjoner på en sånn måte at også brukeren forstod det helt i starten. At brukeren kan være mer engasjert i spesifikasjonene. Sånn som vi har skrevet spesifikasjoner, så har vi skrevet dem i prosa, det blir tolka av utvikler, til å komme som først figurer, og så som skjermbilder. Jeg forstår ikke helt den genovaprosessen likevel, men jeg tror at det kunne vært bedre hvis man hadde hatt et mye bedre samarbeid mellom bruker og utvikler, sånn at de...at bruker forstod prosessen helt ifra starten av. Og slapp det her fjaset med å skrive ting i prosa. For det er noen som slettes ikke kan uttrykke seg klart. Jeg tror faktisk at i den tekniske verden, så er det mange flere dyslektikere enn hva det er som jobber hos NRK, for eksempel. Og det er kanskje helt naturlig. Jeg ser mange dyslektikere som er jævlig flinke. På tekniske ting. Og det er noen brukere som kanskje har litt problemer med å uttrykke seg riktig, når de skal ta sånne tekniske tanker, sette dem på prosa, for at en utvikler skal ta prosaen og oversette det til tekniske tanker igjen. Hvorfor kan de ikke bare bli enige med en gang? På en teknisk tanke. (kassett slutt)

## C.2 Intervju med prosjektleder i utviklerorganisasjonen på TelMont

*Innledning om hvordan intervjuet skal foregå...*

E: Så jeg vet ikke om du kan bare begynne med å forklare litt om hva formålet med prosjektet egentlig var..eller er?

H: Ja, altså det..formålet det er jo egentlig når du skal konvertere fra...til ISDN, så er det..hver gang du har ISDN så har du dobbelt..altså hver gang du bruker ISDN-kort i sentralen så tar det dobbelt så stor plass når du bruker ISDN, slik at du må rydde i sentralen, slik at du på en måte får mer plass. Så det er liksom..kort og så er det plass her og her.

E: Ja.

H: Og så (uklart) sammen, og så er det hull, og så skal du rydde. Rydde på en eller annen måte. Det kalles for et komprimeringsprosjekt. Og det er også videre...tanker om noe som heter kabelomlegging, det har vi begynt...det har man begynt å se på nå. Kan du si. Men vi har ikke fått til noe kontrakt på akkurat den biten så det står litt sånn..akkurat det..Da er det snakk om å flytte..flytte kabelabonnter, altså, (uklart) gi aksessnett i stedet.

E: Mmm.

H: Nå var jeg kanskje ikke helt klar....Skal jeg tegne det opp litt mer, eller noe sånn?

E: Ja, det hadde sikkert vært ålreit.

(H begynner å tegne og forklare rundt tegningen.)

E: Den overordnede arkitektur tilsvarer Wingsprosjektet.... Bare en del sånne praktiske spørsmål, på en måte. Når startet prosjektet?

H: Ja, det spørs hva du mener med starta..Altså, det har jo vært et forprosjekt som har vært kjørt ifra organisasjon A helt ifra, hvis du ser i fjor sommer og før det, så har det vært kjørt forprosjekt som ikke jeg har vært med på, men..Men det var da det ble beslutta å gjøre noe. Men så har det blitt kjørt en ny sånn kan kalle det retningsfase da man finner ut sammen med kunden hva vi skal gjøre.

E: Mmm

H: og den var vel ferdig i september. I september, oktober.Og så begynte man å utarbeide prosjektdirektiv og så videre og så tenke teknisk på hva slags teknologi man skulle bruke og sånn. Da hadde man etablert en kontrakt sånn rett før jul i fjor, da hadde man allerede starta.

E: Mmm

H: Hadde begynt å gjort en god del ting før det.

E: Og den kontrakts...hva skal jeg kalle det...den etableringen av prosjektet med kontraktskrivingen og sånn, var det her, på en måte et offentlig, altså var det andre aktører som var inne i bildet her, var det budgivning og sånn, eller?

H: Nei. Altså, det er ikke et kjempestort prosjekt det her.

E: Nei.

H: Slik at det var ikke det.

E: Nei.

H: Det finne andre prosjekter i organisasjon A som er mye større enn det her. Og de har gjort på den måten som du sier.

E: Ja. Nettopp. Men kravspesifikasjonen, var den noe som var, på en måte, ferdig når dere var på det forprosjektet, eller?

H: Det fantes..på basis av det første forprosjektet før sommeren i fjor, så fantes det en sånn analyserapport, og så lagde vi og organisasjon A en kravspesifikasjon, sammen nærmest. Eller det var jo de som hadde ansvaret for spekken, men vi hjalp til å lage den. Kan du si, omtrent fram til jul.

E: Ja, nettopp.

H: Og så etter det så hadde det også vært mange runder med å detaljere ting og lage detaljspekken. Men akkurat den prosessen der har tatt veldig lang tid.

E: Ja. Har det vært veldig sånn fossefallaktig.

H: Nei. (veldig bestemt)

E: Nei.

H: egentlig ikke, mer iterativt.

E: Ja, nettopp. Har dere laget prototyper og sånt, eller, har det vært mer sånn ut-

vikling på dokumentnivå, eller har dere lagd løsninger for å evaluere aspekter ved. H: altså, vi lagde jo en prototype for å finne om teknologien holdt mål, det gjorde vi, så da ble vi ferdig et par måneder seinere enn vi trodde. Men vi har ikke i og for seg endra spesifikasjonene i og for seg på basis av den prototypen. Det er mer noe som går på organisasjon A sin, altså, arbeidsprosess. Og hva disse skal jobbe med...TelMont skal gjøre. Og hvordan de skal jobbe, ikke sant, den arbeidsprosessen for det har ikke vi med noe å gjøre (uklart)..Da må det være design (dette var uklart)

E: Eh.jeg veit ikke om jeg forstod helt det du mente nå. Altså, den prototypen, formålet med den var å evaluere..

H: Det var å bevise at konseptet vi hadde satt opp rent teknisk holdt.

E: Så rent teknisk, ja. Teknologien...

H: ...den fungerte.

E: Men det var ikke en evaluering av arbeidsmetoder og sånn i tillegg? Eller var det det?

H: Nei.

E: Nei, nettopp. OK, eh..Vi har vel allerede kommet frem til hvem som var kunde, og litt om prosjektetablingen, eh..er det fastpris eller timebasert eller hvordan er det det...

H: Timebasert.

E: Ja, det er time for time, ja. (..) Hvilke firmaer har deltatt i utviklingen av det systemet her så langt?

H: Nei, organisasjon B og Genera og selvfølgelig organisasjon A.

E: Ja.

H: På sin side..altså i tillegg så har IDT og Timko på driftsida vært inne, selvfølgelig, men IDT ble sparka ut og så tok organisasjon A over driften sjøl.

E: Ja vel, de gjorde det, ja. Ja, er det noen interessante ting å lære av det, eller er det av ting som egentlig ikke skyldes...

H: Nei, det skyldes ikke det vi har gjort. Det er egentlig organisasjon A internt, så det er ikke så mye vits i å...

E: Ja, nettopp.

H: Det er ikke så mye vits i å si noe om det.

E: Nei, OK. Skal vi se..Var det da faktisk altså utviklere fra organisasjon A organisasjon A eller fungerte de kun som en kunde?

H: De fungerte kun som en kunde, og på den måten at det har spesialister som kan arbeidsprosessen for å definere den i detalj, ikke sant, og være med å spesifisere skjermbilder og teste ut situasjoner og den type.. E: Nettopp.

H: Så det har ikke vært programmerere fra organisasjon A.

E: Nei, nettopp. OK. Hvordan vil du si at fordelingen av ansvar har vært i prosjektet? Det er et veldig generelt og vanskelig spørsmål, et stort spørsmål, jeg vet ikke..Kan du si noe om..Hvordan var fordelingen av ansvar i prosjektet? Har det vært...Hvor mange prosjektledere har det vært, hvor mange...

H: Nei, det er jo ikke så stort, da.

E: Nei.

H: Det er maks en 10-15 personer.

E: Sånn totalt i utviklingen?

H: Ja, det kan ha vært flere inne på veldig kort tid, altså.. slik at det er jeg som har vært prosjektleder helt fram til det ble ferdig nå. B har overtatt for neste fase

som vi er i nå. Men vi får se om det blir en annen eller ikke.

E: Det vet dere ikke..

H: altså, det er jo fordi det er kontraktsforhandlinger og sånn.

E: Ja, nettopp.

H: Så vi kan ikke si for mye om det..

E: Nei, nei, selvfølgelig. OK, nei, nettopp. Ja vel, så det er i hvert fall da, du kan si...det er veldig inkrementelt sånn sett, da. Fra kunden sin side. At kontrakten også er per fase eller per inkrement eller hva man skal kalle det. Hvor da neste fase er nye forhandlinger i det spille her nå?

H: Ja, det er det.

E: Ja, nettopp. En annen ting som jeg lurte litt på er, litt generelt om prosjektet, hvem er det som egentlig skal vedlikeholde systemet i etterkant? Er det...

H: Det er organisasjon B.

E: Ja, nettopp. Det var egentlig det som var sånn generelt om prosjektet. Så vidt jeg kan se..Ja. Så har vi da litt spørsmål rundt kunderelasjon, da. Nå er det litt spesielt her, kanskje, i og med at dere er jo...dere sitter jo tett på hverandre sånn som jeg har forstått, da. Organisasjon B og organisasjon A organisasjon A, sånn på mange måter.

H: Ja, men altså, ikke helt. Det har vært andre prosjekter der dem har jobbet enda tettere.

E: Ja vel.

H: enda tettere sammen vil jeg påstå. Det har vært prosjekter som har vært kjørt der dem hele tida har sittet i de samme lokalene og jobbet. Det har man ikke gjort i det tilfellet her. De har bare vært innom her sånn i perioder, spesielt og forbindelse med testing og spesifikasjoner.

E: Ja vel. Så både litt i begynnelsen og litt i slutten, på en måte?

H: Det har jo vært jevnt over da, men spesielt i testperioder og sånn veldig mye.

E: Ja. Akseptansetest..har dere hatt en akseptansetype protokoll som de har liksom vært med å kjøre og...

H: Felles system var avtalt (uklart, men tror det var det.)

E: Så egentlig, slik jeg oppfatter det du sier, så har de vært med en kunde enn de har vært på en måte en samarbeidspartner? Altså, de har på en måte vært inne for å evaluere, men altså, føler du at de har...eh..har dere hatt god kontakt med dem, eller har de...har de vært villige til å samarbeide om andre ting enn å bare teste om systemet har blitt slik de har spesifisert? Eller har det vært mange..en del sånne ting, da?

H: Nei, altså..Det har jo vært en god del diskusjoner om ting fram og tilbake, spesielt når framdriftsplaner sprakk av forskjellige årsaker.

E: Mmm.

H: Slik at altså, det har jo vært en dialog hele tida, som ikke alltid har vær like god.

E: Ja, nettopp, ja..(litt småprat om konfidensialitet) Har kunden vært med på å evaluere...altså, vi har snakke litt om test her, men har de vært med på design? Har de vært med på den biten?

H: Ikke av tekniske løsninger, men de har jo vært med på detaljdesign av skjerm-bilder og sett...vært mye inne der, og vært med på å modellerere arbeidsprosessene, eller de tre forskjellige...

E: JA, du mener arbeidsprosessene som produktet på en måte..

H: ...følger.

E: Ja, nettopp, nettopp. Eh...Hvordan synes du samarbeidet med andre utviklere i...nå var jo ikke du direkte utvikler, da. Samarbeidet med prosjektmedarbeidere i prosjektet, hvordan synes klimaet, eller på en måte kulturen har vært? Har det vært bra?

H: Nja, jeg synes det har vært bra. Altså, jeg synes...det har vært litt sånn preg av at det har vært problemer i perioder med å få tak i nok ressurser. Det gjelder både internt og eksternt. Vi har slitt..

E: Mmm.

H: ..en god del akkurat med det.

E: Hva slags ressurser er det dere har hatt..

H: Nei, utviklerressurser som kan akkurat den teknologien og plattformen. Slik at jeg tror ikke man har klart å få helt hundre prosent treff, altså. Hvis man har vært heldig så hadde vi kanskje hatt en person som hadde kunnet det aller meste, men det hadde vi ikke. Det har bare vært med personer som kan deler...så vi har uansett måttet vært nødt til å lære opp folk.

E: Ja, så det har kanskje vært litt vanskelig å ha...har dere hatt en, på en måte, en overordnet arkitekt eller designer på systemet som har prøvd å holde i alle trådene? Er det kanskje deg, eller?

H: Nei, egentlig ikke, for jeg har mest gjort prosjektadministrative ting. Og det er jo Knut som har vært inne sammen med B.

E: Nettopp. På det overordnede..

H: På det som har med design å gjøre. Jeg har ikke gjort så mye med det.

E: Nei. Jeg lurte på om du kunne prøve å beskrive, la oss si, de viktige rollene i prosjektet, eller arbeidsoppgavene, om det var noen du synes kanskje var spesielt viktige på det prosjektet her, eller om det var noen du synes var litt overflødige? Eller var det noen roller eller aktiviteter du savnet?

H: Jeg vet ikke om det er egentlig noe vi har savnet. Jeg kan jo si litt om hva slags roller vi har hatt.

E: Ja.

H: Altså, det har jo vært altså de som jobba en god del med design, og så har det vært de som har jobba mer med ren programmering, sånn som K og B. Og en som heter Jon Erik. Det har også vært et par andre. Inne fra Genera så har det vært et testteam, og organisasjon B har et eget testteam, et eget opplegg som bruker sitt eget verktøy, og de har også vært inne i forbindelse med å forberede testing..og lage testplaner og gjennomføre test.

E: Ja vel, så det har vært et eget testteam, ja.

H: Ja. I organisasjon B. Det har ikke...altså ikke et eget testteam for TelMont, for den delen, men det er...for vært prosjekt så går de inn i prosjektet og så bistår med det som trengs. Både integrasjonstest, systemtest skråstrekk akseptansetest.

E: Ja vel.

H: Normalt sett så er det jo egentlig kunden som har ansvaret for akseptansetesten, men pga den modellen som har vært her så har det vært litt mer flytende.

E: Mmm.Nettopp. eh.. Så det er egentlig ikke noen roller du har savnet? Det er ikke noen spesielle aktiviteter eller....altså...a la prosjekter du har jobba i tidligere så har du kanskje hatt en egen....jeg vet ikke....person som har vært ansvarlig for konfigurasjonsstyring og satt retningslinjer for det. Har dere hatt en sånn person, eller?

H: Nei. Det er såpass lite prosjekt.

E: Ja, så det er noe som hver enkelt har styrt med?

H: Du kan si atte ting som jeg har savna litt, det går mer på oppdragsgivere og styring fra oppdragsgivere. Den har ikke vært helt god. For det har vært to veldig sterke personer her. Prosjektlederen og en til som til dels har vært uenige.

E: Mmm.

H: altså, det har gjort ting litt vanskelige til tider.

E: Ja, har de hatt forskjellige oppfatninger av hva systemet skal gjøre egentlig, eller? Er det det som er...

H: Nei, ikke av hva systemet skal gjøre, men forskjellige oppfatninger av måter å gjøre ting på. Man har mye diskusjoner rundt....den ene sier det ene på ett tidspunkt og så kommer den andre og sier det motsatte.

E: Ja, når du sier måten å gjøre ting på, mener du da reint teknisk eller mener du mer sånn....arbeidsprosjektgjennomføringen?

H: Ja, prosjektgjennomføringen og sånn hos organisasjon A og en del sånne ting og det med hvordan de skal bistå med testing og ressurser og en god del sånne ting har det vært litt sånn krøll med.....

E: Ja, nettopp. Det der er jo litt interessant, for det er en veldig tilsvarende erfaring som jeg har da etter å ha evaluert det Wingsprosjektet som jeg snakka om, hvor også på en måte kunden bidro med testfasiliteter der, og det medført en del problemer.

H: De har ikke bistått med fasiliteter, men de har bistått og hjulpet oss med å teste, ikke sant.

E: Ja. Nettopp. Og det har vær litt uenighet rundt tid og ressurser som skal brukes på det og.....

H: Ja, både på ressurser som vi trenger og på å gjøre detaljdesign og testing og....sånne ting. Altså, vi har jo nærmest.....vi har jo nærmest krevd at de skal komme opp med så og så mye ressurser. Altså, så vi har jo fått det til slutt, da.....

E: Ja.

H: men du kan si, altså, spesielt i en tidlig fase av prosjektet så følte vi at vi hadde ikke nok fokus i organisasjon A, at vi fikk ikke inn fagpersoner som kunne arbeidsprosessen tidlig nok. Og det har gjort at spesifikasjoner og sånn har endra seg ganske mye etter den første kravspekken, altså. (uklart) og det er litt sånn uheldig. Da ryker tiden fort.

E: Ja, ja, nettopp. Er det slik at den som var initiativtaker fra organisasjon A sin side kanskje var en annen person enn da den som endte med å være ansvarlig for prosjektet fra organisasjon A sin side?

H: Nei, det er ikke sånn, men...Det er ikke sånn. For det har vært den samme personen som har vært ansvarlig hele tida. Men det har vært sånn at vi har ikke fått inn.....altså, fagekspertise tidnok som kan fortelle oss i detalj hvordan arbeidsprosessen er og hvordan de jobber, og derfor har det vært litt vanskelig å designe situasjonen i detalj. Altså, hvordan skjermbildet ser ut og hvordan overgangen mellom....altså hva som skal skje i detalj.....

E: Ja, nettopp. Men når dere har fått tilgang til de ressursene, så har dere prøvd å lage da prototyper eller prøvd å ha ganske god kontakt med de tidlig, da....

H: Ja, altså, fra etter jul, så har det vært i og for seg bra kontakt.

E: Ja, nettopp. OK.

H: Da synes jeg det har vært bra, men det som har vært før jul det har vært litt for tynt faglig sett.

E: OK. Hvor viktig har det bidraget der....altså, det å beskrive arbeidsprosessene må jo være en svært viktig del av kravspesifikasjonen her, altså....

H: Ja, uten den så lager du et system som sannsynligvis ikke vil treffe helt, ikke sant. Og da vil jo ikke organisasjon A få den nytten av det her greiene som de har fått nå, så altså, det er jo helt vesentlig for at det skal bli et vellykka prosjekt.

E: Ja, nettopp. Mmm. Hvis du skal gi en sann.....Jammen, det der var veldig nyttig. Bare for å få sagt det først. Nå begynner vi snart å gå litt mer teknisk. Men jeg tenkte på, kan du si noe om omtrentlig.....det er jo veldig vanskelig spørsmål, altså, dere hadde jo en rekke aktiviteter, design og analyse, kravspesifikasjon, prototyping og ha det måtte være for noe. Testing og så videre.....Kan du si noe om på en måte, den fordelingen av ressurser, tid, kostnader, de forskjellige aktivitetene?

H: Ja, jeg har absolutt alt inne i et regneark, men jeg tror det er bedre at vi etter lunsj tar og ser på det enn at jeg driver og forteller det her.

E: Ja. kjempebra.

H: Jeg har også i og for seg gjort en måling. Av prosjektet. Altså vi har estimeringsteknikker og sann, som da har vært brukt, og da estimeres det i forkant, så det er klart at det måler hvor effektiv du har vært i forhold til det i etterkant, og begge deler har vi da gjort. Og det er delvis brukt en metode som kalles for funksjonspoengestimering.....Og der har vi jo tall på forhånd på hvor mange timer vi skulle ha brukt og tall på hvor mye vi har brukt og hvor effektive vi har vært i forhold til det vi trodde og en del sånne ting, da.

E: Ja, nettopp. Hvordan har dere.....det er litt interessant å vite. Dere har jo jobba med andre webprosjekter tidligere.... H: Ja. Ja, men ikke oss som gruppe her.

E: Nei.

H: Så det må jo sies at det har vært veldig mye kompetanse som har vært bygd oppi det prosjektet her. Og det er klart.....det vi har erfart, er at det tar ganske mye tid. Mer tid enn de vi trodd, egentlig, på forhånd.

E: Men den funksjonspoengberegningen deres, den krever jo vanligvis.....altså, det å telle funksjonspoeng, det kan jo være en vanskelig jobb i seg sjøl, en annen ting er jo at du må helst ha en.....en slags modell som sier noe om forholdet mellom tidsforbruk og funksjonspoeng, da. Og den krever jo litt....Hvis det her er et nytt prosjekt, så....

H: Altså, vi brukte erfaringstall fra tidligere webprosjekter som vi har hatt, men dessverre så viste det seg etter hvert at de prosjektene ikke var så relevant som det vi trodde. Fordi at det ligger mye mer forretningslogikk inni det her enn det de ligger inni andre systemer som var mer rene presentasjonssystemer pluss litt logikk. Og spesielt på COM/MTS-sida. Og det gjorde at vi bomma, rett og slett. Og det ganske stygt.

E: Ja, nettopp. Dere undervurderte på en måte den transaksjonshåndteringen og den businesslogikken som ligger i det mellomlaget der.

H: Ja, litt fordi at vi.....altså, jeg trodde at de andre prosjektene, at de hadde hatt mer businesslogikk enn det de viste seg å ha.

E: Ja.

H: Fordi.....så hadde vi da tatt utgangspunkt i dem og justert noe, men vi estimerte tydeligvis ikke på langt langt nær nok.

E: Det der er jo også et annet veldig interessant tema som jeg også er ganske involvert i da, i en del andre bedrifter, men det er jo en annen sak....Eh....ja, skal vi se....Ja, vi er jo snart i mål, vi. Prosessen, altså.....overall, hvordan vil du klassifisere den utviklingsprosessen dere har brukt for prosjektet, altså, fossefall, iterativ-inkrementell, prototypingsprosjekt, hvordan vil du klassifisere det?

H: Vi har jo ikke brukt Genovaprosessen, da, men vi har jo fra før noe som heter

løsningsleveranseprosessen, som er veldig sånn iterativ på en måte, at en skal jobbe underveis og underveis...at kunden skal bli enige om ting og så utvikler vi litt og så bli enige om mer og så utvikle....det var i utgangspunktet en sånn modell som vi hadde tenkt på. Men den måtte vi slutte med. Etter hvert. Etter hvert så måtte vi bli mer og mer formelle.

E: Ja vel.

H: Kan du si. Og å ha en mer prosess der man hadde veldig klart godkjenning av spekker og så programmering og sånn. Men meninga var...tanken min var at man ikke skulle følge noen sånn fossefallsprosess til å starte med. Det har litt med kontraktsformen og, at den er timebasert eller ikke....og ikke fastpris.

E: Ja.

H: Og at man har sett tidligere i organisasjon A at sånn typiske fossefallsmodeller det blir veldig rigid og dyrt sånn fort. I forhold til der man har en mer iterativ prosess som har bedre kontakt med kundene.....

E: Så da er det opplagte neste spørsmålet er da: hvorfor dere da etter hvert gikk over til fossefall? Eller noe litt mer formelt, som du sier, da. Hva skyldtes det?

H: Altså, et skyldtes at etter hvert så kom det...det kom endringer hele tida.....

E: Ja.

H: Veldig seint.....endringer endringer endringer.....Og så for å bli ferdige så hadde vi blitt nødt til å stoppe det der, ikke sant.....

E: Nettopp. Det der er veldig interessant.

H: Rett og slett for å få økonomi innunder kontroll og tidsplaner og alt mulig.

E: Det er veldig interessant. Så det du sier er at det er veldig bra å jobbe iterativt, opp til et punkt hvor man på en måte bare må bestemme seg for at nok er nok, nå har vi surra nok, nå er det på tide å lage det vi har bestemt oss for å lage. Det høres jo nesten litt sånn ut, da, at.....

H: Ja, men det skal sies det at vi hadde problemer med selve teknologien og den prototypen også som ikke har noe med det å gjøre samtidig, så begge deler påvirker jo...påvirker jo også hverandre, ikke sant.

E: Ja. Men var det slik at du....følte du press...eh.....utenfra, som gjorde at du ikke følte at du kunne forsvare å jobbe iterativt?

H: Nei, vi kunne ikke forsvare å jobbe på den måte der. Omlegginga ble vel gjort sånn i februar-mars.....

E: Mmm. Nettopp.

H: Måten den ble styrt på, da. Det her blei jo diskutert med oppdragsgiver, da, selvfølgelig. Måten det ble gjort på, det ble gjort noen endringer i prosjektdirektivet og....

E: Ja.

H: For å få ting mer formelt.

E: Ja. Var det dere som initierte den endringen til mer formelt, eller var det ....

H: Det var vi som gjorde det på basis av tilbakemeldinger. Altså, jeg har kjørt en del prosjekter før, som har vært veldig uformelle og det har gått bra, men det er klart, hvis det begynner å bli spørsmål om framdrift på økonomi og dels sånn veldig mye fram og tilbake som det da ble, så er du da nødt til å stramme inn, og de andre prosjektene jeg har kjørt, dem...noen av dem har vært nesten forskningsprosjekter, ikke sant, så da er det....er det liksom noe annet.

E: Mmm.

H: Da jobber du bare utover og du får levert....levert en gitt funksjonalitet etter en gitt tid, ikke sant, da har du bare levert det, og der har det ikke vært så sterke



krav til hva man skal levere.....Det er blitt mye mer tydelig i organisasjon A de siste åra, at det har blitt veldig mye mer sånn, strengt kunde-leverandør-forhold. Det har litt å si for hvordan prosjektet er drevet, da.

E: Ja.

H: Jeg tror den løsningsleveransemodellen var laga i en tid der vi mer skulle være den interne IT-avdelinga. Til resten av organisasjon A, men jeg ser helt klart at den måten å jobbe på kanskje ikke er....er så lur hvis vi blir sett som en ekstern leverandør av.....Men det har veldig mye å si også hvordan de tenker på oppdragsgiversida.

E: Ja, det var jo veldig interessant, da. Slik at da....ta Genovaprosessen generelt, eller kanskje med webtilpasning spesielt, uten at det nødvendigvis er det viktige poenget her, om det er med webutviklingsprosjekter eller annet, så får jeg jo litt inntrykk av at det du sier, er at det ofte kan, litt avhengig av oppdragsgivers holdning til det, så kan en sånn iterativ utviklingsprosess kanskje ikke være så god pga krav til på en måte prosjektoppfølgning og kontrakter og formaliteter rundt det.

H: Ja, det er riktig. Altså, måten som man egentlig tenker å løse det på når en tenker løsningsleveranse er jo det at man har iterativt, men man har relativt korte tidsbokser, med tre måneders varighet eller noe sånt, men problemet er at vi måtte komme oss opp på et visst nivå rent teknisk og få fram en plattform, og det tar betydelig mer enn tre måneder, så det er liksom første....

E: Mmm. Jaja, nettopp.

H: Problemene ble at det ble litt vel langt, selv om det var planlagt at det var milepæler og sånn slik at man skulle dele det opp, på en måte, i en prototype som man hadde fått da etter, ja, to-tre måneder eller to måneder eller noe sånt.

E: Ja, to-tre måneder etter...fra hvilket punkt, egentlig?

H: Fra man starta å utvikle i oktober, november.

E: Ja, nettopp. Så den første leveransen var på en måte en prototype etter to-tre måneder, ja.

H: Ja, nå brukte vi jo lenger tid enn vi skulle, da.

E: Ja, nettopp på grunn av, som du sier, at dere måtte ha opp en plattform som tok.....

H: Ja.

E: Ja, nettopp. Så du kjørte da iterativt fram til, hva var det du sa, fram til.....?

H: Til mars.

E: Ja, nettopp. Og da.....Så etter det har det vært mer krav, design, kode....test.

H: Det er klart, vi hadde en kravspekk som baserte seg på.....som vi trodde var god nok. Men så viste det seg da at det var den ikke, og så sklei det ut, og så måtte vi til slutt gjeninnføre at den detaljspekken skal være ferdig og godkjent til da, og så programmere ferdig den, og så kommer det en nye en, og så....

E: Ja, nettopp. Men når du sier at den sklei ut, var det pga...var det kunden som kom med nye eller endrede krav, da, eller var det at dere var usikre på kravene, eller var det en kombinasjon eller hvordan vil du, på en måte, analysere det?

H: Nei, altså, etter hvert som man satte seg mer ned sammen med de fagekspertene så fant man jo ut nye ting og hvordan ting kunne forbedres, og hvordan ting måtte være litt annerledes, ikke sant, og du kan jo si at på basis av det så ble det jo hele tiden nye krav og endringer i krav.

E: Men hadde systemet på en måte, i gåsetegn, "fungert" hvis vi ikke hadde hatt den interaksjonen med kunden?

H: Nei. Det hadde jo fungert rent teknisk, men ikke sånn som du skulle bruke det.

E: Nei, nettopp. Så dere hadde nok iterasjoner til dere fant ut at, OK, nå er vi på en måte på et sted hvor vi føler at....at vi vet hva systemet skal gjøre, og så kjørte dere mer fossefall?

H: Ja. Det kan du jo si. Men nå er vi vel....altså, systemet ble vel en to-tre måneder forsinka i forhold til det som opprinnelig var tenkt.

E: Mmm.

H: Slik at det er jo bare siste biten som jeg følte var litt mer sånn....Men jeg tror kanskje vi burde ha starta det enda før. Muligens. Men problemet vårt var det at det var ikke så helt enkelt, heller, å gjøre ting før....altså, vi måtte få opp en fungerende prototype og bevise at teknologien virka først. Og det tok veldig mye mer tid enn vi trodde.

E: Ja. Mmm. Veldig interessant dette her. Ja. Og så er det noen store spørsmål, da, så da kan du jo svare da, forhåpentligvis, så mye eller lite som du vil på det. Jeg lurte på, hvis du hadde hatt ansvaret for det prosjektet her, på nytt med det du vet nå, på en måte, er det noe du ville ha gjort annerledes hvis du hadde hatt mulighet til å gjøre om igjen? Er det for eksempel spesielle punkter i enten prosess eller prosjektetablering eller teknologi, prototyping, kunderelasjon eller andre ting, som du ville utelatt eller gjort annerledes eller....

H: Jeg ville ha kjørt et mer formelt løp tidligere. Mer formell måte å være på tidligere.

E: Men....OK, når du da sier formell, hva..kan du beskrive hva du legger i det begrepet?

H: Ja, mer det at man skal ha godkjent..godkjent formelle spesifikasjoner tidligere. Også det med å være enda flinkere med å kjøre formelle statusmøter med skriftlig dokumentasjon på hva som er gjort for hver uke og litt sånne ting. Og da faktisk...faktisk få bruke mer tid på administrasjon og kundepleie enn det det blei gjort i en tidlig fase. Tror jeg at jeg ville ha gjort.

E: Ja, nettopp. Så det betyr da at du kanskje ville gått da enda tidligere bort fra en sånn løsningsleveranse- type.....

H: Ja. Det betyr det.

E: Ja. Det er jo veldig interessant. Er det noe spesielt med det prosjektet her som du føler at er årsaken til det? Det har vi kanskje vært litt inne på allerede, da, men....Altså, gjelder det....hvor generell er den oppfatningen du har nå, på en måte? Kan du generaliser det du sier nå ut over dette prosjektet her, eller er det helt spesifikt for dette prosjektet her, at du ville gjort det mer sånn formelt på dette prosjektet, men kanskje ikke generelt?

H: Ja, altså....det blir mer og mer formelt sånn i organisasjon A, for vi blir mer og mer en ekstern leverandør enn en intern IT-avdeling, slik at det vil nok....det vil bli mer og mer sånn, på ting som vi leverer både (uklart)...at vi har nødt til å være mer formelle.

E: Mmm. Så du sier altså at etter din oppfatning....etter din erfaring, så er det å være en ekstern leverandør av programvare, så egner den type iterativt utvikling seg ikke så bra?

H: Da skal en i hvert fall være forsiktig med hvordan en bruker det. Være veldig forsiktig og ha kontroll (banker flere ganger i bordet)....punkter hele tida underveis. Det er det at hvis kunden....hvis man begynner å diskutere med kunden i etterkant, og man får overskridelser på tid og pengeforbruk, så er det da problemet oppstår. Hvis man da har vært veldig uformelle, for da kan jo kunden spørre seg: hva er...hva er...hva er det som vi skal levere i forhold til det som er avtalt levert. Og så videre,

ikke sant. Og så...og så...og så begynner man å diskutere. Hvis man da har hatt en prosess som går sånn hele tida der man gjør endringer hele tida, så er det fryktelig vanskelig å.....

E: Nettopp, nettopp. Mmm.

H: Altså, vi mente jo at en god del av overskridelsene skyldes den måten kunden hadde vært på, hvor man stadig kom med endringer i spesifikasjoner og så videre. Jeg kjørte ganske hardt på det.

E: Ja, selvfølgelig. Men det er kjempeinteressant. Jeg mener at dette her er viktige poenger å få med i en prosessbeskrivelse, en prosessbeskrivelse slik vi ser det skal også inneholde erfaringer rundt: bør du bruke denne prosessen? Det skal ikke være sånn at man skal på en måte selge inn sånn som Rational Software Corporation gjør, som de sier at Rational Unified Process is the process, og den skal du bare bruke på alt du gjør her i verden. Det ønsker vi ikke å gjøre. Altså, dette her er jo veldig interessant. At man skal tenke nøye gjennom valg av prosess, i hvert fall. OK. Eh...vi har egentlig vært inne på en del av detaljene rundt komponentene som systemet består av. Det har vi da også skrevet ned også, så det er det ikke nødvendig å gå noe mer inn på. Eh...dere brukte UML eller noen annen notasjon i designfasen, gjorde dere det?

H: Ja, vi gjorde det. Det var jo meninga å generere kode og sånn, men akkurat den biten har ikke vært spesielt vellykka. Altså, reverse engineering for å gå tilbake i modellen og sånn, det har vi da ikke fått gjort. Altså, vi hadde nok håpa å utnytte verktøyet bedre....

E: Og verktøyet var Rose, eller?

H: Ja, Rose ikke sant. Bedre enn det som de klarte å få til. Men jeg tror det har vært...det har vært veldig nyttig i en startfase å kjøre...og kjøre analyse og design på den måten der. Men nytten av det der etter hvert som prosjektet skred fram, har ikke vært fullt så stort. Fordi at de klarer ikke å holde oppdatert koden og....

E: Og det neste spørsmålet mitt er: er modellen holdt oppdatert med nye leveranse, og det er det da tydeligvis ikke?

H: Nei, og du får tidspress på deg, og det har vært tekniske problemer og sånn å få det der gjennomført, ikke sant, og....

E: Det er også litt spesielt med å jobbe iterativt, da. Hvis du....la oss si at du bruker UML i en fossefallsmodell, så er det jo veldig enkelt å ha modellen konsistent med koden. Du sier jo nå at hvis du virkelig bare koder det du har spesifisert, så er jo den konsistent i en fossefallsutviklingsprosess. Men i en iterativ prosess, så er den jo da ikke det. Og det er jo virkelig en stor jobb og..brukte dere bare en klassemodeller og brukte dere bare UseCases eller hvilke deler av UML var det dere konsentrerte dere om?

H: Klassemodeller, UseCases gjorde vi....

E: Sekvensdiagram?

H: Sekvensdiagram gjorde vi også delvis. Nå er det ikke jeg som har jobba mest med det, så jeg tror det er best du spør....Men vi kjørte også faktisk en slags UseCases ikke ved hjelp av Rational Rose, men på papir, noe som du kaller brunpapirmetoden først. Og akkurat det å gjøre det først og ikke bruke verktøy, det tror jeg var veldig lurt.

E: Javel, ja. Det er interessant. Og den brunpapirmetoden, den er å skrive ned funksjonalitet på masse lapper og sånt, eller....

H: Jeg kan vise deg.

E: Ja, det hadde vært veldig interessant. Vi kan ta det etterpå. Det....det hørtes

veldig interessant ut. Men skal vi se....jeg har fått med meg at dere brukte et slag UseCase, men fulgte dere brunpapirmetoden opp med en formell UseCasemodell?

H: Nja, vi gjorde det, men jeg er usikker på hvor langt, altså, hvor langt i detalj de gikk med.....Det er ikke jeg som har vært med på den delen her. Da er det bedre å snakke med B. Slik at jeg ikke sier noe gærent.

E: Du kan da i hvert fall si det du på en måte vet om det, da, ut i fra ditt synspunkt. Som prosjektleder. Synes du den biten her med UML og UseCases og sånn var nyttig. Brukte dere UseCasemodell en mot kunden, eller innen en del av kravspesifikasjonen mot kunden? Eller var det mer sånn internt her, eller.....?

H: Vi har nok vist det til dem, men jeg tror det var mer slik at vi brukte den her brunpapirmetoden først. Slik at vi har vel ikke brukt den her all verdens mot kunden, det har vært internt.

E: Ja. Ja vel. Da har jeg litt mer tekniske spørsmål, da. Eh.....Har du noen ide om hvor mye tid dere har brukt på å rette opp feil, enten i.....det er jo veldig relatert til mye vi har snakka om da, men enten til...i kravspesifikasjonen eller i design og i kode, og eventuelt da, i hvilke deler, komponenter av koden synes du det har vært mest på en måte feil....eh...feilsensitiv, kan du si?

H: MTS/COM.

E: MTS-koden har vært slitsom?

H: MTS mot COM. Vi har ikke fått den til å fungere helt skikkelig slik den skal mot Oracle. Blant annet også det å generelt bygge COM-objektene. Det kunne vi litt for lite fra før, rett og slett.

E: Men et har da vært mest det tekniske rundt COM og....og og...MTS Microsoft Transaction Server, ikke sant? Det tekniske rundt det mer enn businesslogikken, er det ikke det? Som det har vært mye....altså, de objektene dere hadde der, de implementerte...hvis jeg har forstått det riktig, businessreglene deres?

H: Ja, altså vi hadde store problemer til å starte med med å få prototypen til å fungere. Vi måtte gjøre om ganske mange ting. Og jeg må innrømme at jeg er ikke spesielt fornøyd med Microsoft på en del ting der. Og også det med å få Microsoft til å gå mot Oracle. Begge leverandørene skylder p å hverandre.

E: Ja, nettopp, selvfølgelig. Så det også bidro til at det tok lengre tid enn dere forventa å lage den første, på en måte, plattformen.

H: Altså, hvis jeg hadde visst det jeg hadde visst i dag, så ville jeg nok ikke ha valgt....ville brukt Microsoft database, ikke Oracle.

E: Ja vel, fordi integrasjonen med SQL-server er mye bedre, eller?

H: Ja.

E: Ja.

H: Grunnen til at Oracle ble valgt, var at den gangen så skulle Timko drifte det, og de hadde erfaring med drift av maskiner med Oracle på, så da valgte vi bare Oracle.

E: Ja, nettopp.

H: Men sånn i ettertid, så ser jeg jo det at det kanskje var dumt, på grunn av den....det trøbbelet som vi har fått, på grunn av det.

E: Ja, det er jo interessant om det er et generelt fenomen, dersom man bruker en Oracle database, at man da også burde bruke Oracle mellomlag. De har vel, så vidt jeg husker, fra den tiden jeg var utvikler for ikke så veldig lenge siden, ja, så hadde jo de en rekke mellomlagsløsninger de også, i likhet med MTS.

H: JA, det er godt mulig. Men jeg tror jeg nærmest ville sagt det motsatt. At jeg ville kutta ut Oracle, og brukt Microsoft database, fordi at det her er en såpass liten

applikasjon, og såpass lite data i, at det spiller kanskje ikke så stor rolle akkurat hvilken database man bruker.

E: Nei, det er mer det å få en enkel og integrert teknologi, ja. Som spiller på samme lag.

H: Ja, det er mer det.

E: Nettopp. Mmm. Det er bra. Ja, jeg har også et spørsmål om redesign av systemet. Har dere hatt mye redesign, restrukturering av koden? Eller av systemet, for å si det sånn. Og i tilfelle, hvorfor?

H: Altså, det har vært gjort om en god del ting når det gjelder i forbindelse med.....(kassetten måtte bli snudd)

E: Ja, da er det også noen mer tekniske spørsmål. Men som du sikkert har ditt, du har en viss formening om likevel. Eh....Litt mer sånn spesifikt mot webapplikasjoner, for det første, hva synes du er den største forskjellen mellom å utvikle webapplikasjoner og andre typer systemer? Hva er særegent med å utvikle webapplikasjoner som du ikke ville se dersom du skulle lage en sånn mer tradisjonell tjenerløsning?

H: Det er først og fremst knytta til måten du lager GUI på. Mener jeg. Jeg mener fortsatt at når det gjelder GUI, så er webapplikasjoner betydelig mer begrensa enn om du bruker tradisjonelle verktøy. Og det er også tildels noe tyngre å lage, og du må bruke mer tid på å lage det enn du ville gjort med tradisjonelt verktøy som PowerBuilder eller et eller annet.

E: Mmm. Er det derfor viktig å få GUI på plass, altså, at GUI blir riktig ganske tidlig. Altså, du sier at det koster ganske mye å endre på GUI, med webapplikasjoner.

H: Ja, jeg veit ikke om akkurat det med endringene. Det er generelt tyngre fordi verktøyene ikke er så moderne, mener jeg.

E: Ja vel, nettopp. Og det er den største forskjellen, slik du ser det?

H: Jeg vet ikke om det er....ja. Altså, du blir jo veldig bundet inn i et rammeverk i hva browseren og sånn støtter, da.

E: Ja, nettopp. Sånn teknisk sett, så er det den delen av det. Men hva med prosjektgjennomføringsting. Er det noen forskjeller der. Eller, dere har kanskje ikke hatt noen aktører inne som har vært ansvarlige for design av brukergrensesnitt og at dere har måttet forholde dere til så veldig mange andre firmaer sånn sett?

H: Nei.

E: Nei, nettopp. Dere har gjort alt innad (uklart).....

H: Ja. Mmm.

H: Altså, det er egentlig en relativt tradisjonell applikasjon som vi har implementert på en webplattform. Slik at vi har ikke..vi har ikke utnytta en del av de egenskapene i web.....som man kunne gjort i andre....som man kunne brukt i, utnytta mer i andre sammenhenger. Grunnen til at det ble lag, hadde noe med tanke på fleksibilitet å gjøre, og en del sånn ting å gjøre, at man slapp å installere ting på....Og også sånn fordi oppdragsgiver ville ha applikasjonen sånn.

E: Ja, nettopp. En del sånn da rundt....rundt "look-and-feel" og brukergrensesnitt og sånn. eh.....har dere lagt mye vekt på det, at det skal estetisk være, at det skal være en elegant løsning, har dere brukt mye tid på det, eller? At den skal være lett å lære for brukerne, eller?

H: Altså, vi har brukt ganske mye tid på design og detaljdesign, og brukt på brukergrensesnittet, ja.

E: Ja, nettopp.

H: Det har vi. Men også det at vi tok det....det er et sånt rammeverk rundt appli-

kasjonen som vi har henta, eller gjenbrukt, ifra andre webapplikasjoner. Altså, et rammeverk bygd på farger og ramma rundt og også pålogging.

E: Ja, altså. Virkelig et rammeverk som er...er det mer retningslinjer a la desingpattens, eller er det kode som er gjenbrukt?

H: Kode som er gjenbrukt.

E: Nettopp. Det er objektorientert kode det er snakk om, eller?

H: Jeg vet ikke hvor objektorientert det er. Ærlig talt.

E: Men er det Java eller er det....

H: Nei, det er jo Microsoft.

E: Ja, ASP og....

H: Ja, det er det.

E: Åja, OK. Ja, men det er interessant. Utformingen av grensesnittet på applikasjonen, så var det tydeligvis da et samarbeid med, i veldig høy grad, et samarbeid med organisasjon A organisasjon A?

H: Ja.

E: Ja, nettopp. Og hvem fra organisasjon A sin side, hvem var det da som var med på design? Var det virkelige brukere, eller var det.....altså, de som virkelig de som til slutt skal bruke applikasjonen eller var det eksperter fra organisasjon A organisasjon A sin side som vet hvordan.....

H: Til å starte med var det med mer eksperter. Dessverre. Men etter hvert så har vi fått inn noen som virkelig er brukere av det, som virkelig har peiling. Fra vi fikk inn det, ble det veldig mye mer.....

E: Det var kanskje også derfor dere hadde en del endringer i kravspekken?

H: Ja. Det samarbeidet der har vært veldig bra etter hvert. Det virker som at selv om vi brukte lenger tid og sånn, og har gått over budsjettet og sånn, så er de veldig fornøyde nå. Noe skal sies, at vi har også hatt trøbbel på andre sida. Altså, ikke med det vi har gjort, men med det som de system teknisk (uklart) har levert til oss. Som også har kosta oss ganske mye hodebry, spesielt i testfasen til slutt.

E: Så det har vært en utfordring her, og integrere på en måte et system, eller lage et system som integrerer så mange forskjellige komponenter, da, både stormaskin, eksisterende stormaskinsystemer og Oracle og webklienter og hele den derre greia der?

H: Ja. Og så har det vært sånn at vi har spekker som går mot de to andre systemene, og så har vi levert til dem, og så har vi fått tilbake nye spekker, og har gått etter det, og så har ikke de helt levert det de har lovt.

Det har vært mye frem og tilbake, for dette her...Og det er ting som ikke vi kan påvirke fordi disse andre prosjektene også har andre ting de skal ha implementert i de systemene, og så har det blitt ressurskonflikter og prioriteringskonflikter og....Det er noe som heter PLT som prioriterer .../AS, og der er også organisasjon A inni bildet, det påvirker oss, ikke sant, det har også påvirket oss en god del. Det er sånne utenforliggende ting som kommer inn. Og som gjør at man forbruker enda mer tid. Men det er typisk det, når det er flere parter inn i bildet, så blir det gjerne sånn.

E: Ja, nettopp. Ja, vi har i grunnen vær gjennom det meste av de andre spørsmålene. Dere har jo gjenbrukt komponenter fra andre webutviklingsprosjekter, og..et spørsmål, her, lagde dere et navigasjonskart? I tillegg til detaljtegning av...spesifikasjon av brukergrensesnittet, lagde dere et navigasjonskart på en måte, som viste hvordan man skulle navigere gjennom GUI-en. For å implementere de forskjellige delene av funksjonaliteten i systemet? Altså, nå har vi lagd sånne flytskjemaer, rimelige greie

sammen med organisasjon A, som viser hvordan man skal gå igjennom, det har vi lagd, men jeg tror ikke vi gjorde det i UML og sånn.

E: Nei, nei, men dere gjorde det uformelt.....

H: Ja, det har vært diskutert mye. Og har endra seg en del.

E: Ja, og hva er det som har vært diskutert mye?

H: Ja, altså....Du har jo bokser som sier at vi du gjør det, så skjer det, og så videre, og det har jo endra seg litt hvordan man skal gjøre det underveis. Og da har vi jo også i og for seg endra en del i etterkant når en har satt opp et nytt trinn. Da har vi gått gjennom og satt opp nye sånne løyper for.....

E: Og det er rimelig kostbart egentlig også? Altså, dere har jo tydeligvis brukt ganske mye tid på å endre det der navigasjonskartet, og brukt det som kommunikasjon med kunden, da?

H: Ja, det har vært brukt en del tid på det, men det er jo....Altså, det er jo....akkurat det er det jo kunden som egentlig har hatt ansvaret for. Akkurat det.

E: Ja vel.

H: Akkurat det skjemaet, så har vi bare vært med. Som en rådgiver, kan du si.

E: Mmm. Ja, at de har på en måte....at det har vært de som har kommet med det som et slags kravspesifikasjon, eller som en del av en kravspesifikasjon? Så er det dere som har sagt: er dette her teknisk mulig, eller en god løsning på en måte?

H: Ja, det er omtrent sånn.

E: Yes. Nå har jeg snart snakka nok, med mindre du tror det er noe spesielt jeg har glemmt, eller utelatt, eller ikke tenker på?

A: Nei. Jeg kan spørre om det med sikkerhet. Har du...har dere hatt problemer med sikkerhet, tenkt mye på det, eller?

H: Altså, alt ligger inne på TID-nettet, slik at det ble jo....det har vært vurdert en god del. Og om man skal ha enda større sikkerhet uten å ha en sånn passordløsning og sånn. Men i trinn 1 så ble det der kutta ut. Det var meningen at de skulle komme sikkerhetsløsninger i trinn 1, men det ble kutta ut på grunn av tidspress, og på grunn av at man kanskje anså at det man gjorde i trinn 1 ikke var så kritisk fordi at man er jo innenfor organisasjon As nett hele tiden. I trinn 2 nå, så er det snakk om å gå ut, og da er det jo noe helt annet. Da stiller det helt andre krav til sikkerhet. Altså, det som i og for seg er farlig med applikasjonen her, er at det er ikke dataene i seg sjøl, om noen får se dem, som er så viktig, men det er hva man kan gjøre ved hjelp av systemet. Altså, hvis folk begynner å rote til en sentral, ikke sant, på den måten, så kan det være helkrise.

E: Jada.

H: Og spesielt på seinere trinn, der man kan gjøre enda mye mer. Med sentraler og sånn....hvis man gjør feil, så kan man sette helt ut av spill en sentral i Oslo med 40 000 faste abonnenter, ikke sant.....

E: Ja, nettopp. Ja, for det derre AXE-systemet, det er vel....det er et gammelt sånn, det er nettverksadministrasjon er...(veldig uklart)

H: Nei, altså, AXE er selve sentralen...leverandøren til sentralen som inneholder sånn...det pluss systemet rundt, ikke sant.

E: Ja vel, OK. For vi sender jo kommandoer ned til sentralen, da. Som blant annet driver og flytter ting der. Det er klart, hvis du gjør ting gærent og ikke har skapt det...da blir det kaos. Altså, de som driver med systemet her, de er fagekspertter som har peiling på det. De skal gjøre....det er en relativt begrensa brukergruppe på 50 personer, eller noe sånn.

E: Mmm. Åja, det er ikke større. Det er på en måte brukergruppen på det systemet?

H: Her på trinn 1, så er det det. På trinn 2 så vil det nok.....hvis det blir noe av det, så er det nok betydelig flere, slik at det er en begrensa brukergruppe. Som da er eksperter.

E: Nettopp. Mmm. Er det noe mer du....som sagt, formålet her er å høste erfaringer for å definere, komme med innspill til en forbedret prosess, som skal inneholde en del retningslinjer for hvordan man skal kunne lage sånne løsninger som det her. Og det er jo helt klart at de tingene du har snakket om så langt, det tror jeg absolutt har vært veldig nyttig. Jeg lurte på om det er noe mer du kan.....som du tror kan være relevant som ikke vi har tenkt på i de spørsmålene vi har gitt deg så langt? Hvis du skulle utforme en Genovaprosess for webutviklingsprosjekter, hva ville du lagt spesielt vekt på og spesifisere eller definere i en sånn prosess? Aktiviteter, eller leveranser, eller andre råd. Kontraktskriving eller?

H: Nei, altså, det er jo å være enda mer nøye i en tidlig fase.

E: Ja.

H: Det gjør veldig mye av prosjektet.

E: Mmm. Mmm.

H: jeg synes i og for seg.....det har vært...har vært vellykka det at man har brukt verktøyet til Rational Rose og modellert. Det har vært vellykka, men vi har jo ikke fått fullt så mye ut av det om det vi trodde.

E: Nei.

H: Altså, vi trodde ganske mye i utgangspunktet.

E: Hva slags effekt er det egentlig du synes dere har fått ut av det? Er det...altså, bevisstgjøring av design, eller.....det høres ikke ut som om dere har brukt det så veldig mye til kommunikasjon av løsninger, da, i og med at dere ikke har holdt det up to date. Har det vært mer sånn for hver enkelt å finne ut av hvordan man skal lage løsningen sin, eller....?

H: det har jo vært for å designe løsninga til å starte med. Til å starte med tror jeg en hadde veldig mye nytte av det.

E: Ja vel.

H: Men etter hvert.....Altså, siden man ikke fikk kodegenereringa og reverse engineering-greia til å fungere, så forsvinner en del, det er jo åpenbart at den del av vitsen....den forsvinner.

E: Ja. Vet du hvorfor dere ikke fikk det til å fungere?

H: Nei. I og med at jeg ikke har jobba med det, så....

E: Nei, vel. De lover mer enn de kan holde noen ganger, de på Rational. Nei, jeg vet ikke om vi har noe mer da, jeg.....

A: Nei.

(Avslutter)

### C.3 Intervju med utvikleren "B" på TelMont

E: Når var de forskjellige leveransene, når ble de forskjellige tingene gjort fra starten av prosjektet? Greier du å tegne et sånt oversiktskart sånn at vi greier å lage et slag mentalt bilde av hvordan dette prosjektet her har foregått, når var prototypen, når hadde dere kontakt med kunden og alle disse tingene her?

B: Tidsfasene i det, da. Ja jeg kunne kanskje prøvd å sagt littegrann om det, ja. Sånn sirkadatoer og sånn...

Aller først hadde vi et slags forprosjekt, da, vi startet med et forprosjekt. Kunden



hadde tatt kontakt med oss og vurderte behovet rett og slett. Vi hadde en person herfra og en tre-fire personer (uklart) som ble kalt inn fra dem sjøl også, for å vurdere hvordan ting kunne skje.

Det var vel på tidlig på våren en gang, starta vi, tenker jeg, fra mai 99.

E: Ja, fra mai 99, ja.

B: Ja, mmm..Kunden hadde også hatt noe forprosjekt tidligere, altså, mer begynt å vurdere hva som kunne trekkes inn og sånt noe, men..jeg kan ikke si helt eksakt i lengden..jeg ville tro at de hadde holdt på noe..eh..ja..noe sånt som et halvt års tid, rett og slett....Vi startet i hvert fall vårt forprosjekt våren 99. Og tidlig på høsten da satte man ned kravspesifikasjoner...analyse og kravspesifikasjoner. Det var da to til tre personer fra oss, da, som var med...

E: Hvor mange var med fra organisasjon A?

B: De var vel fire..fire personer. Det er vanskelig å si prosentsetningen. De hadde jo prosjektlederen som var 100 %, og så hadde de to til fem personer ved siden av, en person som var så vidt innom..i den kravspekkfasen

E: I den fasen der så var det prosjektleder etablert både på..

B: Både hos oss og dem, ja.

E: Var det en som var overordnet ansvarlig for koordineringen?

B: Det er klart at...Uansett var vi leid inn som konsulenter, ikke sant, i den fasen her. Det var ikke vi som skulle skrive kravspekken, det var dem som skulle skrive kravspekken..ihvertfall..skal vi skrive kravspekken må vi liksom ta hatten dems..slik at de hadde en prosjektleder som klart tok ansvar for kravspekken ..vi definerte oss mer som konsulenter da, men vi hadde jo også en prosjektleder hos oss som skulle styre det, men vi ble mer konsulentaktige..

E: ja, skjønner...

B: det trakk ganske mye ut det der...det var for så vidt en komplisert affære.

E: Ja, det her var fra august, ja?

B: Ja fra august. Vi stoppa midt på sommeren...i juni-juli, og så kom vi i gang igjen i august, og da holdt vi på i tre-fire måneder med kravspekkfasen. Og klart, da snakka vi jo om de overordnede fasene her, og det var liksom komprimering, kabellegging og speiling?... Det var liksom det som var store bolker her..kravspekken tok jo for seg alle disse tingene..det er klart, det var jo egentlig et ganske stort omfang på den. Så ble det da satt opp prioriteringer og så ble det satt opp en analyse da. Det vi manglet for kravspekken det var da...da begynte vi å etterlyse brukerne, da. De som kom til å bruke systemet. Vi begynte jo prototypinga også, da, i september. Bare en enkel prototype for å vise skjermbilder og for å diskutere rundt det.

E: Det var en del av analysen?

B: Ja, jeg ville definert det som en del av analysen. Analyse..og det er klart, du begynner jo litt design, da. Men vi prøvde å få ut tidligst mulig, da, et skjermbilde så de hadde noe mer håndfast å forholde seg til

E: ja...

B: det var et slit, ikke sant. I kravspekkfasen så blir det veldig...de brukte jo typisk sånne svære gråpapir, og klæsja inn forskjellige ting på dem...det er ikke så lett å forstå dems arbeidsprosess og måten de gjorde ting ut fra og (uklart) ..da prøvde vi å forstå den og tegne opp et skjermbilde ut i fra det. Det gjorde vi i alle disse fasene, eller alle disse tre typene prosjekter som vi..

E: den prototypen som kom ut en gang, da, den ..

B: det var 10.januar, tenker jeg. Da hadde vi en workshop der vi kunne gå

gjennom skjermbildene, gå gjennom arbeidsflyten, gjennom ..vi måtte gå gjennom arbeidsflyten for å se hvordan skjermbilder vi skulle ha. Så vi kunne se problemstillingene....men fortsatt hadde vi ikke...vi var ute i organisasjonen og hadde veldig problemer med avklaringen her. Det forsinka oss egentlig. De drev og fant på ting, ikke sant, altså...

E: På det tidspunktet her hadde dere altså ikke reelle brukere?

B: Vi hadde vel en egentlig, som var reell...men likevel, det var jo...klart, det var flere typer prosjekter, ikke sant og det var ...de engasjerte seg jo i forskjellig grad i prosjektet. Vi hadde jo en som blant annet var med...i kravspekken. Og var veldig fokusert på noe som heter kabel(.uklart).og det var jo ingen av de andre..altså han, var ingen bruker lenger. Han hadde vært en bruker en gang i tida...

E: ja.

B: og for så vidt vært en initiativtaker til at man..fordi dette her var jo store gevinster å spare, ikke sant.

Han hadde tatt et stort tak der i noen års tid, ropt høyt, for å si at nå måtte man snart komme med noe her, det var mye å tjene på et sånt system..

E: ja, mmmm

B: men han var nå da satt til andre oppgaver, mer sånn..

E: ja

B: han var mer sånn IT-rådgiver

E: ja, nettopp.

B: så han var mer...han var med og stilte veldig store krav til systemet

E: Ja, nettopp.. Flexibilitet og?

B: Ja, veldig mye sånn...

E: Så han var riktig sånn stakeholder og initiativtaker på det prosjektet her egentlig?

B: ja, egentlig var han initiativtaker og ønska veldig stor grad av funksjonalitet på systemet.

E: ja. Selv om han da egentlig ikke var en bruker?

B: ikke bruker og heller ikke noen økonomisk ansvarlig..

E: nei, nettopp.

B: For kostnaden...

E: han var ildsjel, altså, som...

B: ja, han var en ildsjel...

E: som mente at her var det kostnadsbesparelser etter som han hadde sett utifra den erfaringen han hadde gjort og hadde fra tidligere?

B: Riktig

E: Interessant.

B: Så vi så jo han som bruker i utgangspunktet, ikke sant, men vi innså mer og mer at han var jo ikke den brukeren vi egentlig var interessert i. Vi definerte oss jo en bruker....

E: Så har han egentlig solgt inn dette her hos ledelsen sin igjen, da? Var det sånn det egentlig hadde skjedd eller var det litt..?

B: ja, det var ikke...det var vel egentlig ikke han som bli systemeier, han jobber jo da..er nå ute. Tidligere hadde han vært bruker av systemet. Eieren av systemet som økte effektiviteten ...som så..skulle ut og bli brukt av brukere, ikke sant..

E: Ja, ja.

B: Det var trinn midt imellom der og de som bestilte systemet som egentlig skulle bruke det.

E: Ja. Kan det forklare da..Så vidt jeg har forstått, nå etter intervju tre her, så er det jo...så har det vært et problem å få tak i de reelle brukerne.

B: Ja, det var et problem. Jeg vet ikke ...Ikke før i februar hadde vi kontroll over den situasjonen, da hadde vi endelig disse reelle brukerne.

E: I februar, ja..

B: Februar, vil jeg si.I januar, i slutten av januar, begynnelsen av februar, kan en si. En kan si overgangen januar-februar. Jeg er ikke helt sikker nå, altså...

E: Mmm..

B: Jeg antar det er noe vi egentlig ville hatt i november, vi hadde jo veldig mye sånn...Prøvde å få skalert det opp og sånn...Da fikk vi jo også brukere da, som ikke hadde så mye sånn ansvarsforhold overfor kravspekken.

E: Mmm...Nettopp.

B: Og som kanskje også ei lita periode ...Noen av dem brukte kanskje et par måneder, kanskje, før de ble varme nok til å protestere på kravspesifikasjoner. Altså, man..typiske svar man fikk, ikke sant, jo..men, ja, de turte ikke si nei, altså, rett og slett, til alle de kravene som kanskje ikke behøvde å ligge i systemet.

E: Men etter hvert så gjorde de det?

B: Ja, de gjorde det ..ja, de kom etter hvert, ikke sant, man fikk jo skrelet bort en del ting etter hvert i den derre..i prototypinga forsåvidt..men selv mens vi holdt på med prototypinga så var de redde for å si at, at, fordi jeg ikke visste nok om bakgrunnen, ikke sant, så tørde ikke de å si at, nei, det er ikke noe vi ser for oss.

E: Ja.

B: Så fortsatt lå det for mye kompleksitet i systemet som vi brukte tid og krefter på

E: Mmm.

B: Så det var mer kanskje at brukerne var litt ...fremmed for kravspekken. Jeg skal være litt forsiktig her rett og slett, gå litt forsiktig fram , for det kan hende at det er noen andre som tenker annerledes ikke sant.

E: Ja, mmm..

B: Og så snakket vi typisk om, om tre forskjellige prosjekttyper. Og det var spesielt egentlig veldig mye han som var pådriveren her som kom med..med mange ting som for så vidt kanskje ikke vi følte som ...(uklart på bånd)

E: De har spesifisert ganske..litt mer..større system enn det man kanskje...

B: Ja, det også, men..men det var jo det man begynte..vi ville jo snevre det inn. (uklart)..Arbeidsprosessene er veldig kompliserte. Det trengte nå..vi hadde jobba...det er også et problem, at dem hadde jobba forskjellige steder. De hadde forskjellige rutiner for ting, det hadde jo ikke vært noe system tidligere

E: Nei

B: så da hadde de tilpasset det her og tilpasset det der hele tiden og det var kanskje også en litt grunn til at de ikke turte å si nei, helt sånn, nei, det ..jeg kjenner liksom min region, det...hvordan vi jobber, men jeg vet jo ikke hvordan de gjør det i Oslo, ikke sant.

E: Mm.

B: De mangler...de visste ikke helt åssen.

E: Var det dere som hadde ansvaret for å prøve å koordinere ansvaret for disse forskjellige lagene av ...

B: De samlet, de samlet dem da typisk på disse her workshopene vi hadde, da, vi hadde jo .(uklart)kravspekken, så...disse her gråpapirene, og forstå hva de ville, ikke sant, de var samla i noen workshoper hvor vi satt og diskuterte og ...kom med

innspill og...og sånne ting. Men vi hadde ikke brukerne i den første tiden, ikke de reelle brukerne, selv om vi trodde jo nesten at de var brukerne...

E: Ja.

B: Man sitter også tror at det..og det samme gjorde vi med prototyping...vi hadde gjennomgang. Da hadde vi hele dager som vi gikk igjennom en versjon av prototypen og så tok den ned og fikk tilbakemeldinger og så kjørte vi en ny runde litt senere.

E: Med oppdatert..?

B Med oppdatert ..

E: Hvor mange ganger tror du dere gjorde det med oppdateringa av prototyper?

B: Jeg tror vi hadde tre sånne større...tre sånne store...tre dager, da, som vi hadde, som vi kjørte gjennom hele..

E: Mm. I en workshop, ja, med mange forskjellige...

B: Vi tok folk inn her, ikke sant.

E: Fra over alt?

B: Og brukte minst en til to dager, brukte vi.

E: så det var ganske nyttig, da?

B: Det er klart at vi fikk jo..fikk jo lagt punkt på de kravene som egentlig var viktige for dem, og vi fikk jo fram , i diskusjonene at, ting var mindre viktige som kanskje egentlig skulle....(mumling) etter hvert da. Vi fikk en prioritering på hvilke arbeidsprosesser som skulle være først, det kom egentlig fram her

E: Ja

B: Men poenget her er at: det kan vi vente med, det kan vi vente med, vi har innsett det, at det . Det var komprimeringa som var det letteste for oss. Og å klare å følge arbeidsprosessen, da,

E: Men da virker det nesten som at...det at dere tross alt da hadde flere prototyper og sånn, så betyr det at dere egentlig her.. så var det dere som hadde ansvaret for...eller i hvert fall en veldig viktig del..i utarbeidelsen av en analyse/kravspek, kan du si...

B: det hadde det vært hele tida, ikke sant, vi hadde jo kjempe..altså, vi var jo de som kanskje hadde tid til å jobbe med kravspekken og, vi angra litt på det i etterkant. At vi hadde vært ...kanskje ivra så mye som vi hadde gjort, at vi egentlig burde satt oss litt mer tilbake. Eller, egentlig ikke tatt så mye ansvar, kanskje spesielt ikke prøvd å skrive ned kravspekken. Klart de hadde jo problemer med å strekke til med ressursene sine. De hadde jo andre dager og gjøremål, mens vi..vi tenkte at vi skulle gjøre det vi var betalt for å gjøre og ..

E: Nei, jeg begynner å få en sånn viss formening om at...altså, å jobbe iterativt, inkrementelt, med prototyping liksom sånn litt eksplorativt...måte å jobbe på, skyver jo på en måte det tradisjonelle ansvaret som kunden har for å lage en kravspek over på leverandøren av produktet. Fordi at du da tross alt skal levere en prototype, og så blir det da forhandlinger, reforhandlinger og så videre av kravspekken?

B: Vi hadde jo også litt sånn...vi prøvde å si at ..vi prøvde jo også å være...(uklart) detaljspekker.

E: Mmm.

B: som liksom skulle være en oppsummering, da, egentlig. Og vi var jo viss på at det var de som skulle lære seg kravspekken. Det begynte jo å bli så stort sprik fra den opprinnelige kravspekken.

E: Mmm

B: Og det som da ble de detaljspekkene det var jo..(Uklart).. FORDI det hadde vært en iterativ prosess.

E: mm.

B: Men som da ble litt lang fordi man ikke hadde brukeren inne.

E: Nettopp.

B: Vi begynte å tenke litt enklere med en gang. Vi tenkte nesten med en gang produkt allerede her. Vi begynte automatisk å tenke på begrensningene og løsningsvalg. ...hvordan løsningene skulle bli. Det kan jo være både riktig og galt det egentlig, men..sånn er det jo

E: Mmm

B: I denne fasen her...Samtidig så begynte man å gjøre prototyper , folk var jo med her en hel måned og implementerte skjermbilder og sånne ting. Og det var jo også ...funksjonaliteten bak ble jo også kjørt her i..

E: En annen ting: Den prototypen ble jo også brukt til å ..egentlig evaluere teknologien etter som jeg har skjønt.

B: Ja, vi kjørte en mer sånn teknologiprototype, da.

E: Var det en annen prototype?

B: Eh...ja, altså..ja, det vil jeg si. Jeg vil kalle det en annen prototype. Første runden var det for så vidt det samme skjermbildet, det var de skjermbildene som var viktige i den andre prototypen, da.

E: Ja, okei. Så du vil si..du hadde på en måte en teknologiprototype, hvor dere testa ut MTS og alt dette her?

B: det ble også kjørt.også kjørt i januar.

E: JA, nettopp.

B: Vi startet i midten av desember.

E: Så den andre prototypen, det var altså en samling av skjermbilder, var det..var det funksjonalitet implementert holdt jeg å si...?

B: Altså, den første prototypen vi hadde i januar, det var den samme altså, vi så den som en helhet

E: JA, nettopp

B: Klart vi gjorde det.

E: Ja, mm.

B: Men det er klart vi hadde ny workshop på skjermbildene (uklart)for å få fram det egentlig i skjermbildene

E: Ja. Okei, men bare sånn, litt sånn for å...Så dere brukte da prototypen både til å evaluere teknologi og egentlig til å klarlegge krav og til å kjøre workshop?

B: Ja, det gjorde vi.

E: Mmm.

B: Vi startet jo egentlig en implementasjon da. Den prototypen var mer en ..en implementasjon da. I slutten av desember så ble den starta. Dermed så ble dette kjørt og brukt i desember. Men det er klart at det lå ikke under det, altså, det var ikke noe ..det ble ikke knyttet så mye funksjonalitet...

E: Mmm

B: for jeg vil jo si at den prototypen man hadde som basis den ble jo mye tyngre enn det man egentlig hadde forutsett her.

E: Ja

B: Kanskje den ene feilen var at vi la inn ny funksjonalitet igjen..

E: Ja, nettopp. Men altså, den teknoprototypen, den begynte i november, desember

B: Jeg vil nå si januar, jeg. Men...

E: Ja vel  
B: men det var egentlig helt i slutten av desember  
E: Ja vel  
B: Hvis jeg husker riktig nå  
E: Ja, så det var her ja. Nettopp.  
B: vi ønsket å få en liten vurdering da, , jeg vil si at vi ønska det i februar en gang..vi ønsket å få sagt at: her er den pakka vi satser på, og nå har vi fått noe som skulle gjennom hele..(uklart)  
E: Apropos det...Var...før du sikkert har mye mer å si om det, men..eh..apropos det å evaluere da teknologien ved hjelp av en sånn prototyp – hva slag strategi brukte dere da for å ...prøvde dere å implementere en veldig smal..liksom vertikal...eller horisontal prototype da på en måte?  
B: Vi hadde jo plukka ut noen skjermbilder, ikke sant, som vi hadde sett for oss, og det var..altså..eh..kanskje vi ikke hadde bestemt oss bra nok her. Sånn jeg ser det gikk vi for bredt ut  
E: Mmm  
B: man slet for mye faktisk med det som ikke skulle implementeres .(uklart) det dro veldig ut i tid.  
E: Akkurat  
B: kanskje litt for vanskelige deler av applikasjonen. Jeg husker nå hvor mange timer vi satt og diskuterte litt att og fram hvor man ser på tegningen..er det klart i forhold til hva dem egentlig ønsker seg her. Det var kanskje en litt kompleks sak vi tok tak i. Vi burde hatt mindre omfang, altså. Det ble for mange metoder man skulle kjøre gjennom.  
E: Ja, nettopp. Dere hadde altså litt problemer med at den prototypen den var ikke veldig fokus..det virket som om dere hadde flere mål med den  
B: Ja, det ene var jo teknologien, det andre var jo det at man tross alt skulle lage profesjonelt også, ikke sant, man burde kanskje ...  
E: dere hadde ingen planer om å kaste den prototypen? Den var...  
B: Nei. Det..nei.  
E: det ville ikke kunden heller ha gått med på, går jeg utifra, eller?  
B: Det er vanskelig å si...vanskelig å si.  
E: Ja. Hva synes du nå i etterkant, da, burde dere lagd en litt mere rå prototype som dere bare hadde hivd etterpå, eller?  
B: Jeg tror det. Jeg er overbevist om det, ja. Vi burde ikke tenkt så mye på at..at vi skulle få den til å kjøre. Uten å tenke så mye på teknologien, så lenge det var en ny teknologi (Uklart). Og det tok litt lang tid også før vi tok inn hele aspektet.  
A: Hvis dere hadde lagd en mer rå prototype, tror du dere hadde forandra teknologi, da?  
B: Ja, det kan være en stor mulighet for det, ja. Ja det...ja, det hadde vært kanskje litt mere typisk kanskje bedre. Vi hadde jo også en diskusjon om det. Men den diskusjonene var kanskje uti mars eller noe sånt no. da hadde vi lagd så mye funksjonalitet allerede, så jeg vet at de som for så vidt var mest aktive..  
E: Mmm  
B: i prototypinga protesterte veldig for å bytte over til noe annet  
E: Ja. Jaja, men hva synes du om det?  
B: nei, det er jo vanskelig når en først har lagt inn veldig mye tid og lagt inn veldig mye jobb i funksjonaliteten, da er det veldig vanskelig å bare kaste det  
E: Mmm

B: Jeg tror nok at de som..altså, da begynte vi liksom å komme så langt i modelleringa at det var nesten ikke noe mer å skrive, altså vi tok ikke...vi burde egentlig tatt den teknologiavgjørelsen .Vi burde ikke latt den prototypen få lov til å skli ut så mye Man burde innsett at her burde man innsnevret den prototypen

E: Ja.

B: altså, det er greit at når man starter, at man ikke veit helt hvor komplekst dette blir, men når man først hadde holdt på en stund, så burde man sagt at : nei, nå stopper vi med den her og konsentrerer oss om teknologien, men ..(uklart)

E: Det virker jo som om det har vært litt høyt risikoprojekt med hensyn på teknologi da?

B: Ja, og det var jo ..vi burde begynt i begynnelsen, tenker jeg, at vi egentlig var ute etter å få etablert en ny kompetanse innen teknologi i vårt miljø, da. Og da var det liksom: ja, hvordan gjør vi det...Da var det, det er ikke noe midler for å kjøre små piloter, ikke sant, så diskuterte vi sånn...jeg veit ikke, jeg...og sa det at: går vi inn her, så må vi også ha noe kompetanse. Og det er for så vidt det som ble sluttresultatet også...at at vi fikk bygget opp kompetanse . Vi så at: vi får kanskje betalt for det, ikke sant

E: Jaja

B: så vi visste jo at det var høy risiko, det visste vi fra begynnelsen

E: Ja, mm

B: og så hadde vi assosiert oss med kompetansen vår også, da (? Mye bakgrunnsstøy) . som også forsvant litt...noen personer som havna som rådgivere blant annet i den teknologiprototypen, som sa dette er bra, og dette bør vi (veldig uklart her)vi hadde kompetanse på deler av det

E: Mmm

B: Om ikke annet

E: Mmm

B: vi hadde også litt problemer på grunn av Oracle database og Microsoftsaken vår. Ting som tok veldig mye tid. Det er ikke så mange andre som har gjort disse tingene E: Det slår meg at web...å utvikle webapplikasjoner er faktisk litt annerledes enn å utvikle andre applikasjoner, mer tradisjonelle applikasjoner, i hvert fall nå, med..den tilstanden ting er i nå? Resonnementet mitt er bare det at det virker som om..at.at det er veldig mye ny teknologi.

B: det er det. Det er . Men du kan jo spørre om det alltid vil være sånn, men du ser jo nå hva som rører seg

E: Jeg tenker bare sånn..formålet her er å definere en prosess for webutviklingsprosjekter. Det virker på meg som om en nyttig er faring her er at..eller i hvert fall da for..kan du si...et eget kapittel for første gang man lager en webapplikasjon, husk på: og så vider, og der kan vi jo ha en god del underpunkter som man kan trekke erfaringer blant annet fra dette prosjektet og en rekke andre prosjekter vi skal ..som ser ut til å ha veldig mye til felles, i hvert fall på det at det er mye teknologi å sette seg inn i

B: Ja, absolutt. Teknologien går så fort at det er mye nytt for mange, at det er veldig mye..det er noen flik her og noen der og så har man forsåvidt..man får ofte tilbakemelding på at problemet går bra, ikke sant, og så har man får tilbakemeldinger, må forhøre seg litt, ta noen telefoner og sjekke ut ting...hva slags problemer kan vi støte på

E: Mmm

B: og da er det ofte lite kompetanseutprøving. (??)man har at ..jo..man kan si at i

teorien går det bra

E: Mmm

B: ikke sant, men faktisk, i det praktiske, så er det vanskelig å få det til å gjennomføre...altså, det er tungt, ting henger ikke sammen. Det er kanskje noen som har gjort litt for enkelt forsøk på tingene, og så viser det seg at man må bruke..at skal man lykkes i et prosjekt så er det...er det mye tøffere.

E: JA, mm

B: du lærer veldig mye, men det er likevel en tung reise. Tung vei å gå, altså.

E:Ja, mm.

B: jeg ser bare at det andre alternativet, det blir å diskutere nå (uklart. En utvidelse?) og det er jo et sånt helt annet, og det er jo sånn det var...for det første så var det jo ..vi visste vel kanskje litt om det på våren en gang..så var det prat om i hvertfall litt fjernstyringsalternativ via internett eller sånn som en utvidelse. Og det det viser seg nå er jo at der stiller man helt andre krav til teknologi..til bruk og til å mikse andre mulige arkitekturer da inni der. ..altså, da kan man si at man kanskje burde gå over til...Java beans og CORBA og Unix og sånn altså, man kan si at det er konkurrenten på en måte..så da har man for så vidt en oppdragsgiver som .som har en annen plattform for det igjen, som da har dukka opp og som for så vidt er nokså ny, med nye produkter..det er .ware fra IBM, ikke sant, sånn..det var kanskje i vår en gang en første gang hørte ordet ikke sant, og nå er det plutselig det store, så det er vanskelig å...altså, man har problemer med å med å ha..altså..og problemer med å få tak på de folka som kan det..det er alltid noen som kan det, ikke sant, noen som har klart å være i forkant, noen personer, og de er veldig etterspurt og de er det vanskelig å..de får du aldri inn i prosjektet, de er liksom rådgivere...litt på siden. Det var det vi også hadde her, ikke sant, noen som var rådgivere sånn.

E: Altså også med hensyn på ressurspersoner, å få tilgang på de personene som...

B: Ja, det er også spørsmål om...prosjektlederen. Det er det første du må gjøre, det er alltid i begynnelsen

E: Jo , selvfølgelig, det er det uansett, men det virker jo som om det er kanskje spesielt kritisk ..spesielt kritisk her, da.

B: Det er fordi at det skjer så mye forandringer og at det skjer mye.. E: det skjer mye og det skjer raskt. Nå. Altså, selvfølgelig, sammenligner man klient-tjener, vanlig klient- tjenerløsninger med Powerbuilder eller C++ eller et eller annet sånt for bare for tre år siden, eller sånn.

B: Ja,

E: Men det er kjempe..veldig bra poeng.

B: Altså man skal åpne..altså, det skjer mye forandringer, altså, fra det å være lokal applikasjon til det å være tilgjengelig på mange steder. Det blir mer og mer åpenhet, utviklingsprosessen altså, det er jo det jeg snakker om

E: Mmm

B: som da krever sine løsninger, sine sikkerhetsløsninger, sine

E: Mmm

B: .(uklart) ekstremt vanskelig på en måte, sånn jeg ser det nå, så er sikkerhetsløsninger en..en sånn, det er en skikkelig krevende sak. Og det er nesten ingen som har..man må sikke fingeren i jorda og si at: hvordan kan vi integrere alle disse tingene, ny teknologi med CORBA, med ...det er veldig mye å holde orden på her...

E: JA, jada, du kjenner ikke til noen som har prøvd å lage..designguidelines,



liksom, rundt sikkerhet og sesjoner og..og forskjellige aspekter av ..av...

B: Altså, organisasjon A har jo veldig mye sånn portaler. De begynner å lage sikkerhetsportaler nå for å støtte nye mindre prosjekter, da. (uklart) til forskjellige typer applikasjoner....noen som er nesten helt tilsvarende og noen fleksiløsninger, det er liksom tre sånne typer nå, men det er for så vidt noe som skal støtte de andre. Man klarer ikke å implementere en egen sikkerhetsløsning noen steder, man må ..man har ikke ressurser til det, man har ikke kapasitet til å gjøre alt selv. Så det blir liksom..man prøver for så vidt å lage en slags støttetypeprosjekter da, eller portaler eller hva man kaller det, som skal ta seg av visse ting ved siden av...vi er også i en sånn oppstartfase, da. Innenfor den telekommunikasjonsbransjen, da, så er liksom det Internett-aksess til alt, liksom, og det..typisk, så må man konkurrere om sentraler da, Internett da, det er det jeg snakker om...

E: Ja, ja..

B: Så det er ganske tøft, det.

E: Mmmm...men det er typisk en sånn...komponent om du vil, av en sånn...denne type utviklingsprosjekt som det kan kanskje være en delt gjenbruk i når det gjelder sikkerhetsaspekter

B: Det er klart. Det er jo generelle trekk her, så...

E: Ja.

B: du hadde jo kravspekken her, så var det mange forskjellige faser, så du har flere faser, ikke sant, i et sånt produkt, og da ønsker man jo også å dekke det å gjøre det generelt. Prøve å..(uklart)

E: Men..hvor..hvordan er resten av forløpet her, da, hvis vi går tilbake til den type...

B: Ja, nei, vi var jo...på hæla, da. Så da var det jo mye jobbing og det var en del sånne ..ja, altså, programmeringsfasen den starta altså, ikke sant, med prototypen. Og fortsatte videre nedover. Så var det jobben med å få koordinert...rett og slett...alt som blei..(mumling) og så gikk det da..da begynte folk for så vidt å strammes litt inn og legge planer for hva som gjøres innen de og de ....man bestemte metodene som skulle skrives (veldig uklart)

E: Prosjektstyring på metodenivå?

B: Nærmest på metodenivå, ja. For å...

E: Imponerende

B: Det var Excel, da..excel som vi kryssa av på...

E: Jaja, jojo,

B: Og ga tilbakemeldinger på hvor mange prosent det var gjennomført på den og den og det og det og det...

E: Jaja, mmm.

B: Og det er klart, vi hadde jo brukt UML og sånn, og det...designa en del. Men det er klart, det var jo egentlig å ta utgangspunkt i alle metodene man hadde spesifisert der.

E: Mmm...Den UML-biten, hvor begynte den å eksistere?

B: Den eksisterte nok i desember , den var vel helt i forkant av jula, helt i forkant.

E: Begynte dere å generere noe av prototypen utifra UML, eller?

B: Nei, det var jo et kjempeproblem. Altså, UML og TR-løsningen og sånt noe, det funka ikke i figurasjonen engang.

E: Nei

B: Vi hadde jo egentlig en litt motsatt vei å gå...vi reversa jo tilbake igjen, men det å få det til det var jo for så vidt også litt frustrerende her, under designfasen,

så var det en frustrasjon. Definere metoder, ikke sant, legge inn beskrivelser av hva disse metodene skulle gjøre og sånne ting..så fikk vi ikke det over, ikke sant, vi måtte

E: Ja, fordi dere bare gjorde reverse engineering, ja?

B: Ja, altså vi gjorde ..vi var jo ute etter forward engineering, ikke sant, men det funka ikke.

E: Nei

B: Vi klarte ikke å få det verktøyet il å funke

E: Nei, nei

B: da var det en...nesten et problem(mumling)..frustrasjon var det jo, for det skapte jo koordineringsproblemer,

E: ja. Hadde du nytte av UML? Og Rose? På en skala fra en til ti, liksom.

B: Neeei, jeg vil ikke uttale meg om det, men..det er klart det er en del å klargjøre. Jeg hadde nytte av det. Kanskje den som hadde den største graden av det også, men..

E: Ja

B: det var litt vanskelig i den situasjonen. Det vi gjorde var for så vidt å gå inn i å..legge inn..å definere inn metoden der den var og legge inn kommentarer og greier, og begynne med scenarier og sånne ting (uklart)

E: Mmm

B: Så man..den var tung..jeg vil si at det er en dårlig måte å gjøre det på egentlig. Man går på kodenivå først og så tilbake igjen..og så må man oppdatere ting når man gjør noen forandringer og så var det litt sånn hvor ting ble plassert i koden som ikke var helt...vi ikke var helt fornøyd med.

E: Mmm. Var UML-modellen nyttig for å lage datamodellen? . Og var den nyttig for å lage skjermbildene?

B: Neei, ikke for skjermbildene.

E: Nei

B: Vi bruker nok mere..jeg skulle si...gikk mere over til å konsentrere oss om detaljer per skjermbilde. Og den knytningen fra skjermbildet til metodene.

E: Ja, brukte dere use-case..

B: Ja, det er klart, det var en modning her, altså, det er klart. Det var en modning her.i å bruke UML, blant annet med Use Cases..det er..var det en modning, så var det helt klart på designet. Det var liksom det å vedlikeholde videre, og det å få opp den situasjonen som...(ekstremt uklart, men tror det var mest mumling...)

E: Prototypen...prototypene som dere hadde allerede tidlig, var de tilgjengelige fro disse brukerne? Sånn på egenhånd, eller var det kun når de var her for å se det, liksom, eller var det...fikk de virkelig lov til å leke med de...

B: De fikk lov til å leke med dem..ganske tidlig, ja.

E: Tilgang til..

B: Tilgang til de, ja. Og vi la også ut på noe..sendte noe mail til de...og, planla videre utifra det..til neste gang (uklart....Mumling)Og vi skulle vise dem UML-tegningen og USE Casen også, men det ble på et for høyt nivå for dem, altså...(NB! Dette her var uklart) det fikk dem ikke med seg, det skjønnte de ikke noe av, den UMLen.

E: Mmm..

B: Men altså, det stoppa vi ...vi genererte vel en eller to ganger, tenker jeg, og så stoppa vi. Det var bare kjekt og fint, liksom, det var ..det er ikke, vi var ikke...så lenge vi ikke kunne utnytte det sjøl , så er det klart at brukerne ikke fikk noe ut

av det i det hele tatt..de kunne kanskje få en ide om hvor langt vi hadde kommet, men det var det eneste, altså.

E: Så de kommunikasjonsredskapene eller verktøyene dere brukte mot kunden, organisasjon A, liksom, var vel først og fremst prototypen og de detaljerte skjermbildebeskrivelsene med da også en flyt...

B: Med flytskjema..og, ja, da som vi diskuterte. Vi lova dem et skjermbilde, eller i hvert fall se arbeidsflyten...det var vel først og fremst og se overgangen mellom skjermbilder.

E: Ja, mmm. Var det..var det utgangspunktet for å diskutere..var det utgangspunktet for prototypene, eller, var prototypene utgangspunktet for å lage de spesifikasjonene...

B: Vi hadde en arbeids..en beskrivelse av arbeidsflyten, på en måte,

E: Ja.

B: det var et flytskjema for (uklart). Den var vel en del av...den var vel en del av kravspekken. Så kom krav til analyse, jeg er ikke helt sikker, jeg kan ikke gå nå si i..

E: Neinei...

B: Men...jo, det var nok noen..det var ting der, og det var lagt inn en Use Case faktisk som en del av kravspekken, vi brukte jo UML allerede ..

E: Tidlig?

B: På slutten av ..eller i ..eh...et eller annet sted i kravspekken, vi hadde jo..vi begynte å bruke UML en gang i begynnelsen tenker jeg.

E: Mmm

B: så da tok vi litt med brukeren i Use Case også, da. Det er ikke bare bare å få kunden til å forstå hva Use Casen er for noe, sånne ting, altså, det...

E: Nei.

B: vi synes vel ikke at det..jeg tror ikke at kunden synes at det ga noe i det hele tatt egentlig

E: Nei

B: det som var UML, det...fungerte ikke i det hele tatt sånn. Terminologier og tener og Use Case og ..det er uklart..

E: Så det å drive og..og prøve å..iterativt og i fellesskap og prøve å komme frem til en løsning ved hjelp av den type modeller det er litt sjanseløst?

B: Ja. For oss så var det ikke no..var det ikke ...

E: Nei, nei. Men..

B: Jeg følte vi prøvde litt i starten, liksom, og se om de kunne følge oss på en måte, men det...

E: Nei, nei.

B: Jeg liker ikke..prototypen, skjermbilder, litt flyt, det var det som virka for oss

E: Ja

B: Og så kom at dem faktisk kunne prøve å...etter hvert som vi implementerte, ikke sant, så ble jo implementasjonen tilgjengelig..og det var det...som overbeviste kunden vår. Jeg tror dem var fornøyd med det egentlig

E: Ja. Men etter dere var på en måte...ferdig med de prototypingene og det..det ble liksom, det ble jo systemet, etter som jeg har forstått ..

B: Ja, det ble jo systemet. Det ble systemet, ja..

E: det ble systemet..

B: Og de var jo da med å sjekke opp systemet etter hvert.

E: Så det var ikke egentlig noe da inkrementelle leveranser til kunden, det var mer

rett og slett at de bare fikk tilgang hele tiden, på utviklingsmiljøet?

B: De hadde tilgang hele tiden, ja..vi hadde jo mer og mer versjoner, ikke sant, av disse skjermbildene...

E: Mmm.

B: (Uklart) De satt og så skjermbilder der de satt, og vi også, og så diskuterte vi.

E: Ja, nettopp, så dere la de ut ganske umiddelbart, altså?

B: Ja

E: Og det synes du var greit?

B: Ja, jeg tror det var greit for dem og oss..(mumling) Og pluss at de skulle være med på systemtesten. Og de brukerne som var mest aktive i klareringa, ble også de som skulle utføre systemtesten. Så da kunne de liksom ..de ble veldig opptatt av å se på disse...få tilgang til skjermbildene, fordi at de skulle lage testspesifikasjoner for akseptansetesten, men også da på systemtesten

E: Mmm

B: Så de fikk gjort veldig mye på den fasen. Som nevnt..i etterkant følte de at de brukte veldig mye tid på det.

E: Ja

B: den fasen der..men de har fått over..over (uklart) også over systemet.

E: Mmm

B: De har vært med hele veien

E: Leveransen, på en måte, når var den?

B: men det er klart, det var visse problemer. Det er klart at de klaga på at dette ikke var stabilt nok, men det var spesielt når de begynte å gjøre systemtesten

E: Ja

B: Vi hadde noen menyfeil som plutselig ble uten tilbakemeldinger, så...skjedde det store forandringer på det, og da sleit dem litt, ikke sant...for da hadde de begynt med testspesifikasjoner og sånne ting, og da blei det sånn..

E: Ja

B: vurdering og sånne ting..

E: Ja

B: Vi tok jo den detaljeringen egentlig som vi ble enige om, altså, men likevel så..var det ikke helt bra..så gjorde en seg avhengig ..ganske tidlig av det som var..detaljeringsgraden (uklart). Og vi kunne ikke gå tilbake helt tida for hver detalj og oppdatere kravspekken. Det er en faktor, altså, men jeg tror likevel at de....

E: Ja

B: egentlig var positive. Det var litt sånn ustabilitet nå og da, men..ting som hadde fungert, plutselig ikke fungerte...vi hadde ikke noen problemer med det, ikke sant

E: Neinei, selvfølgelig hadde ikke dere det

B: Men vi visste jo stort sett ofte hvorfor det her..plutselig ikke fungerte. Typisk man...metodene...det var tross alt noen kall, noen skjermbilder

E: Mmm

B: som da ble kanskje oppdatert uka etter

E: Ja. Fra det tidspunktet der...altså fra år 2000, så har dere hatt, virker det som, altså et veldig godt samarbeid, da, med..?

B: Ja, da vi fikk inn brukerne, så..

E:; Da dere fikk inn brukerne

B: så følte vi at...da klarte vi å sammen å snevre inn målet, så det blitt et realistisk mål...men vi slet jo litt med noen av formalitetene her, da ..man har kravspekken,

da...og vi hadde de som da var ledelsen, altså i nett, prosjektledelsen der, de hadde klart ønska seg at denne kravspekken her skulle oppdateres kontinuerlig, ikke sant, så de hadde følt at de hadde kontroll på ...

E: Er den kravspekken oppdatert nå?

B: den ble oppdatert i etterkant, ja...

E Når omtrent da ble den oppdatert?

B: nei, i mai, tenker jeg, eller noe sånt..

E: Ja vel...

B: vi hadde også kravspekk på grensesnittet, og den ble oppdatert raskere..

E: Ja, nettopp, grensesnittet mot de andres s.?

B: Ja, og det var også noen kravspekkdokumenter vi utarbeida, og det var også noe vi utarbeide her. Det var ikke bare kravene til TelMont egentlig, det var også...mye av tida gikk også til kravspekk for XXX (de underliggende systemene for TelMont)

E: Mmm. Det var mer for å forsikre at de tilbød de tjenestene dere trengte for å implementere det dere skulle implementere?

B: Ja, på en måte..vi måtte da ha støtte, ikke sant, for å ....(mumling)

E: Fikk dere den støtten eller ble det mye...

B: Ja, det er klart, vi måtte jo følge..altså, det var jo både en fram..de hadde sin framdrift...det var jo ganske...det var jo..det er store systemer, da..telsus lager store systemer..vi var bare en liten bit av deres leveranse.

E: Første leveranse, det var...?

B: Det var i juni...

E: I juni... Var det et... det var et fullt operasjonelt system, som da ble satt i drift?

B: Ja, det ble satt i drift da, men det er klart at...(uklart)

E: Jaja...

B: Men vi kjørte vel ikke så veldig mange prosjekter gjennom TelMont, men...(uklart + måtte snu kassetten)

(Snupause) E: Det hjelper innimellom å se en kalender, nå har vi hørt veldig mye av det samme her, men jeg tror fortsatt at vi har litt problemer, jeg tror vi må ta ...tror det kan hende at vi kan bruke Knut littegrann på dette her...han har vel litt oversikt også...over et del av de tingene, så kan vi prøve å kvalitetssikre noe av den informasjonen, for jeg tror det bli litt vanskelig for oss å...jeg tror det hadde vært veldig ålreit å gjenskape så mye av prosjektforløpet som mulig. Mest mulig eksakt, da.

B: Ja, riktig,

E: Vi trenger noe å henge disse knaggene på...Men jeg tenkte på litt sånn her....(uklart) grupperte ting.

B: Ja, det er klart at vi...vi...vi grupperte ting...sa at da skulle det gjøres det og hadde tidspunkt for det... vi hadde liksom gruppert det...gruppert det...for de å kunne lettere koordinere...typisk skjermbilder...vi hadde en person som koordinerte ting.

E: Jeg lurte på om vi skal forsette med noen av de andre spørsmålene jeg....

litt småprat\*

...litt rundt prosess og da sånne ting...Jeg..nå har vi spurt de samme tingene så mange ganger at jeg tror vi går litt i surr hva vi har spurt hvem.

A: Men jeg lurte på en ting til deg....Eh...Hva synes du er forskjellen på å utvikle webapplikasjoner og vanlige applikasjoner?

B: Altså...det kan være det med prototyping, den ..men jeg tror at hvis vi hadde kjørt...vet ikke om du kan kalle det mellomløp, jeg...så hadde vi hatt kravspekk-

ene...så det...his du sammenligner så har kravspekken vært veldig viktig...den er liksom alfa og omega liksom for...vi har jo hatt her en veldig sånn...kravspekken har jo ikke vært så forpliktende egentlig, i denne sammenhengen her, da...

A: Mmm

B: Fordi at vi har...altså kunden kom jo inn seint...men også at...at...brukerne kom inn seint og at kunden har vært egentlig enig i at disse detaljspekkene var det som egentlig skulle bli sluttresultatet

A: Mmm

B: slik at vi har egentlig tatt over, kan du si...det vil si at vi fikk...vi fikk en sånn iterativ sak som vi kom oss fram til sluttspesifikasjonen som da var disse detaljspesifikasjonene. Det har vært litt sånn...noen synes det har vært frustrerende...bortsett fra de som er prosjektledere...men en av prosjektlederne synes det har vært veldig frustrerende, han følte at han hadde en kravspek og så viste ikke han hvor langt vi var kommet, og vi...hvor langt vi var kommet med brukerne...

E: Mmm...

B: Han følte at han mista kontroll rett og slett. Selv om han var jo med på noen av disse detaljspekktingene også, men jeg tror at han falt egentlig litt av, han klarte ikke å få...bindi de eksakte kravene, hvilken krav er det som forsvant, så han var jo veldig sånn opptatt av at vi skulle oppdatere kravspesifikasjonen.

E: Han ville egentlig ha den konstant, han?

B: Han ønska..jeg er jo glad ikke vi sa at vi skulle ha en altfor stor(?) kravspesifikasjon, det hadde jo vært en håpløs situasjon..man sitter liksom på to nivåer, og skal drive og oppdatere..

E: Mmm

B: Lage detaljspekkene, og så hele tida..uka etter eller noe sånt..drive og holde kravspekken oppdatert, og det hadde vært et mareritt for den som måtte sitte og gjøre den jobben.

E: Ja

B: for de er liksom..det er to helt forskjellige detaljnivåer, og det var ganske mye sånn...vi hadde jo skissert skjermbildene ganske bra..forslag til hvordan de skulle se ut, litt sånn grovt før designet og sånn, men likevel sånn funksjonelt...

A: Men sånn jeg forstod det på hun Kjersti så forandra dere kravspesifikasjonen ganske mye når dere fikk brukerne inn?

B: Ja.

A: Og det var veldig frustrerende for kunden deres?

Jeg tror nok det at man kan si at han var litt frustrert var at kravspekken ble veldig mye forandra. Jeg følte kanskje at han følte at han mista litt kommunikasjonen mot sine ledere igjen. Han skulle hele tida stå til rette for hvor...hva blir produktet når det er ferdig og sånne ting, og så hadde han problemer med å relatere det til et fast dokument.

E: Hva med prosjektledelsen internt her, og hvordan..var det frustrerende på den måten og?

B: Vi diskuterte en del sånn formaliseringen av de og de detaljspekkene..altså det var jo fordi vi skulle ta en sånn..altså problemet, altså..når er vi ferdige? Ikke sant, når er det du sier at: nå er spekken ferdig? Eller nå er vi helt ferdige, så vi drev jo og kjørte litt sånn at vi skulle ha en formalisme, da, at vi skulle få disse kravspekkene godkjent og fryst. Det ønska for så vidt Reier, eller vår prosjektledelse, da. Han ønska at: nå, nå er vi ferdige.

E: Mmm

A: Mmm

B: Nå kommer det ikke noe mer som betyr noe mer jobb, ikke sant, for det er jo det som man ofte har problemer med, at man sitter liksom i et samarbeid. Men det er klart det var jo organisasjon A som betalte oss egentlig timene her, og da var det jo..da er det jo fort at det kan...Det er klart, de ønsker jo, kunden..brukerne ønsker jo gjerne mer og mer og mer,

E: Men det er ikke de som betaler, på en måte?

B: Nei. Det er ikke direkte de som betaler, nei. Det er de som kan spare på det, eller de som kan få god tæla på det..

E: Men sånn sett måtte dere forholde dere til to egentlig to ganske atskilte partnere her, da. Hvor den ene kunne sitte og prototype sammen med dere i dagevis og ukevis, mens den andre egentlig begynner å tenke penger etter hvert.

B: Ja, vi hadde jo likevel også en person fra prosjektledelsen som også var med på detaljspekkene, og som også var den som skulle signere disse detaljspekkene den gang vi begynte å bestemme oss for å jobbe med dem. Da skulle vi fryse dem, men det viste seg at det fortsatt ble litt forandringer i tingene, men det var mer utefra hva systemet krevde at vi skulle gjøre. Det var lite...mye mindre av den typen det at brukeren kom inn og sa at: "Nei, jeg vil ha det sånn og sånn..."

A: Mm.

B: Så klart, det var jo positivt og negativt med de tingene her, men..Det er klart, det er en annen måte å se det på. Altså, den type prototypeutvikling. Akkurat det at du går iterativt i prosessen, da. Men det kunne godt ha vært en annen type applikasjon også, altså. Det kunne vært...det betyr ikke at det er en webapplikasjon at en skal utvikle på den måten her. Det er ikke noe sånn...det kunne godt ha vært

E: Men på en måte så er den jo spesielt godt egna, da, på en en måte, at du i hvert fall da at du får kunden eller brukeren til å evaluere underveis, så er jo det spesielt enkelt, da, med en sånn webapplikasjon

B: Ja

E: i og med at de...du trenger ikke installere noe på nytt, liksom.

B: det var jo det enkleste av alt, ikke sant, det at en kunne sitte på kontoret sitt og faktisk se hva resultatet var. det var jo derfor vi tenkte at UML skulle være tilgjengelig for dem også, da, at de kunne gå inn og titte.og vi kunne også..typisk sånn testdirectory, ikke sant, at de kunne gå inn å se hvor langt vi var kommet i testinga våres, og vi ville spare oss for veldig mye ny installasjon.

E: Men du er jo litt unikt med web...

B: Ja, du slipper mye dokumentasjon, og så kan du bare ta en telefon, og så kan de gå inn og se..nå ser det sånn ut..og så diskuterte vi noen punkter. Disse detaljene var det jo veldig enkelt å komme seg gjennom. Bare samle seg på et telefonmøte, og så diskutere akkurat den lille saken.

A: Mm

B: Så løste vi det sånn de ville ha det, og så var vi ferdige med den saken.

E: Ja.

B: i stedet for at vi kanskje måtte..

E: For ellers er det jo en ganske stor jobb...

B: sende over et dokument... Der vi hadde klipt ut et skjermbilde eller noe sånt, sendt det over, fått tilbakemelding. Altså da måtte det vært detaljspekk som hele tida måtte...detaljspekk var jo egentlig bare en oppsummering av skjermbilder pluss litt mer ting.

E: Jeg synes jo det høres ut som om dette her er et veldig viktig argument, akkurat

det at det er faktisk noe som er litt spesielt..å prototype og få tilbakemeldinger fra kunden eller brukere. Så webutviklingsprosjekter er faktisk mye enklere å gjøre, når jeg bare tenker meg de type prosjekter jeg har jobba på tidligere selv, der kunden er i Sveits, systemet er typisk i en tjenerløsning i C++, Powerbuilder og Sybase..Drive å sende ned CD-er med svære installasjonsscript og oppdatere databasen som ligger der nede og så videre. Det er en helt annen verden det her, for å kunne få kunden til å teste ut det som er gjort.

B: Nei, her hadde det ikke betydd så mye hvor kunden var hen, han kunne godt ha sittet i Sveits..

E: Ja, her ja.

B: Ja. Men det er jo klart, det er den kontrollen i forhold til kravspek og sånn som ledelsen hadde som er minuset ved det. Det skjer fort for mange forandringer i forhold til å klare å..noen av disse prosjektlederne mister litt kontroll på det, da.

E: Jeg ser jo her at det på en måte..jeg vil ikke si en av konfliktene deres, men i hvert fall vært litt motstridende interesser. Den som initierte prosjektet er ikke nødvendigvis brukeren, og prosjektlederen har ikke heller nødvendigvis hatt samme interesse som brukeren. Og prototypinga har blitt drevet kanskje av det du nevnte med..mest mellom bruker og utvikler, da, dere. Mens det kravet om å få fram og holde på kravspekken har vært mer den prosjektleder..et prosjektlederkrav.

B: Det er klart, det er mye økonomi i det.

E: Ja, nettopp, selvfølgelig.

B: Det er klart, det er jo problemet med prosjektet, at det ble overskridelser, ikke sant, at vi klarte ikke å snevre inn nok. Men..

E: Men brukeren er veldig fornøyd, eller hvordan er det?

B: Ja, det vil jeg tro. Jeg har inntrykk av at de er fornøyde. Vi har hatt lite feilretting, blant annet. Men de feilene som har vært er typisk sånne som man gjør unna ganske raskt. Og endringer som kommer..Det er raskt å gjøre endringer.

E: Så det er ikke blitt helt stabilt ennå?

B: Altså, nei...jeg tror det følger. Altså jeg vil tro at det er større sjanse for å få en god applikasjon ved at man har brukere og de er mer med.

E: Mm

B: de vil jo...de ser jo hvordan de vil jobbe. Og da får du noe som er skreddersydd for dems måte å jobbe på. Til tross for at det er de som ofte vil initiere endringene, men de fikk lov til å være med veldig lenge..det er klart at det ble jo.på et tidspunkt så var vi nødt til å se at: nei, nå må vi ta det som er av endringer...Man kom jo til et sånt tidspunkt der man prøvde å fryse selv detaljspekken..og da ble det jo endringer, da begynte man å notere seg endringer og..og så er det ting man oppdager etter hvert. For det er en veldig..det er en arbeidsprosess som er komplisert. Det er ikke noe som en gjør i ørska, holdt jeg på å si. Det er så mange sånne avarter av unntak, dårlig kvalitet i ITSAF (?) -systemet, man tenker hva gjør en da, hva gjør en i spesialtilfeller hvis de og de datoene mangler? Og det er noen ting som dem har kommet på i etterkant, egentlig, kommet på den type tjenester, den bruksmåten eller sånne begreper, og den passa ikke sammen med det andre, ikke sant, og den burde egentlig ha en egen...definert masse sånne framgangsmåter. Hver eneste tjeneste har sine framgangsmåter....og det er typisk der det kommer inn at de ikke har tenkt på alle mulige ting i utgangspunktet. Dette er en komplisert sak. Og så...en ny versjon nå er typisk at vi kan ha mange ting, da. Vi har ikke fått med nok tid med systemet ((Uklart) og dermed så..er det automatisk blitt utsatt.



E: Jeg har bare et veldig enkelt spørsmål igjen her, jeg, da, og det er: hva ville du gjort annerledes dersom du skulle gjøre prosjektet om igjen?

B: Det er ikke et enkelt spørsmål, nei (latter). Nei, altså, vi er inne på den prototypen, den ville vi nok gjort annerledes i dag. Vi ville nok gjort det enklere..lagd en enklere prototype. Og fokusert på teknologien. Ny teknologi neste gang er klart en...en prototype med ny teknologi. Det har vi diskutert de sikkerhetsløsningene, ikke sant, det er diskutert med kunden. Vi vil nå analysere og kjøre en pilot der, hvis det skulle bli aktuelt. Nå spørs det om det blir aktuelt, da, men...det er noe vi vil holde klart på. At ikke vi kjører en..at vi kjører en enkel sak først.

E: Du ville likevel prøvd å jobbe litt iterativt fortsatt? Eller ville du tenkt deg at på et nytt prosjekt a la det her, så ville du kjørt totalt fossefall?

B: Nei, jeg tenker nok iterativ, jeg tror nok jeg er typen som kan tenke i den banen nå, altså. Jeg tror jeg er glad for at dem hadde sagt kravene i bunn, liksom, blitt enige med...man må jo bli enig med kunden. Klart problemet med iterativ prosess her er at vi har forandret veldig mye underveis, men det har jo noe med forventningene til kunden å gjøre. Altså forventningene som kunden har..I etterkant her ble det jo en del diskusjoner rundt økonomien til prosjektet, fordi at man..vi hadde ikke klart å få..vi tapte økonomisk på det fordi vi ikke hadde dokumentert godt nok, alle disse endringene som kom inn.

E: Mmm

B: På ledernivå, liksom. Vi hadde blitt enige med brukeren: ja, vi gjorde det, men: skulle vi ha gjort det? Det ble mye att og fram-diskusjoner, da, og der..der hadde nok vi..hadde vi forholdt oss mer til kravspekken og sånne ting, så tror jeg vi hadde klart oss. Økonomisk. Så..Man må nok ..vi trenger kanskje et bedre..altså, de som implementerte og brukerne fikk kanskje for stor rolle her, egentlig. Der burde man hatt klarere ...Vi burde hatt klarere sjekkpunkter. Typisk den milepælsplanen. Man burde synliggjort dem mye bedre. Prosjektlederne burde ha fokusert mer på det, og funnet ut hva vi egentlig har fått. Man planla en del sånne milepæler, men man lot dem skli ut. Men man burde egentlig ha (knipser). Nå har vi en milepæl: hva er det vi har her nå, ikke sant?

E: Mmm

B: definert milepæler som vi ikke trengte forandre på. For at ledelsen kunne faktisk se: hvor endte dette her?

E: I aller aller siste liten: du vet jo hva formålet vårt er, alle mulige rare erfaringer som kan bidra til å...som har formål i definering av en bedre prosess? Er det noen andre erfaringer som du har som du vil si hvis du skal definere en prosess, en iterativ inkrementell utviklingsprosess med webstøtte...hvilke erfaringer eller råd er det du ville ha med da i en sånn type prosess?

B: Vanskelig spørsmål. (småprat) uansett..det der å klargjøre med kunden hva man egentlig skal ha, altså. Når man inngår kontrakter eller sånn, på det nivået der, altså. Altså: hva er utgangspunktet? Det tror jeg kanskje er viktigere. Det går på forventninger til..at man har samme forventninger til hva sluttproduktet skal bli.

E: Var dere enige om, egentlig prosessen her fra starten av?

B: det ble diskutert, ja, og det blei...det spørs når du mener fra starten. Vi hadde jo en kontrakt rundt kravspekken, ikke sant, og så ble det jo en ny en ganske seint. Det tok litt tid før man fikk nye kontrakter her. Men da hadde man vel egentlig blitt enige om prototyping og den måten å jobbe på. Uansett hadde man jo ikke de brukerne inne...akkurat i november-desember følte jeg at vi brukte veldig mye

ressurser som vi egentlig ikke burde brukt. Fordi at brukerne ikke var der.....

(slutt på båndet)

## C.4 Intervju med utvikleren “K” på TelMont

E: Hvilken rolle hadde du i prosjektet?

K: Som sagt, altså...databasemodellering. og lage databasen, litt lage prosedyrer, ting som går mot databasen, og ting som går mot MTS / COM: Meldingsorientert, asynkront, MOM .(Uklart ) den biten og Oracle, som sagt, og veldig mye spesifikasjonsarbeid egentlig har jeg vært med på. Så jeg har vært med på den objektorienterte utviklinga, ikke webutviklinga. Jeg har vært med på å programmert litt C++.

E: Ja. Kan du beskrive på en måte hvilke roller de andre deltakerne i prosjektet hadde, litt om..

K: Ja, H har du jo møtt, prosjektleder, og så er det jo B, han har også ..vi brukte jo veldig mye tid i prosjektet på å snakke. Vi var veldig underestimert på sånn detalj ..hva skal jeg si .så vi skulle jo i utgangspunktet kjøre prosjektet uten kravspesifikasjon og med sånn fossefallsmetode, og prototyping, da bruke prototyping for å komme fram til løsningen etter hvert, og det .vi brukte veldig mye tid på det, så B også ..B og meg hadde begge to en rolle som gikk .vi brukte veldig mye tid på å snakke, da. Snakke med organisasjon A Nett om løsningen .Om hvordan løsningen vår skulle bli. Ja, B, i tillegg jobbet han littegrann med COM-utvikling, og ..ja. så hadde vi to hovedpersoner til i prosjektet, en som heter C, som er her i dag også. Han jobba bare med COM. Og en som heter J E som jobba med Web, altså ASP, webserveren vår, da. Og så hadde de noen sanne hjelpere inne, men det er liksom sånn som var basisen i prosjektet.

E & A: Mmm

E: Dere hadde ikke .dere var egentlig mer fordelt på nesten komponenter i systemet, da, enn roller, liksom aktiviteter, dere var ikke delt opp sånn i aktiviteter, det var ikke noen som var hovedansvarlig for design, noen som var ansvarlig for test og noen som var ansvarlig for koding, liksom, det var mer sånn komponentbasert?

K: Til å begynne med, så var jo alle med på design. Vi hadde jo også en testperson til, da, egentlig, vi hadde en sånn testgruppe på huset som tar seg av test. Eh ..Så først så var det det design, og da var det jo .da var vi med på det alle sammen. Men under koding så delte vi opp sånn på .altså, det er en veldig liten overlapp i prosjektet på hvem som kan hva, det er B som sagt som jobber på COM som kan det, og J E med ASP-tinga sine, og jeg med databasene mine, så det ..så vi var delt opp sånn.

E: Gjorde det det vanskelig å kommunisere?

K: Nei, det har jeg ikke oppfatta. Det er et ganske lite prosjekt, så vi vet jo hva vi skal levere. Alle vet hva som skal komme ut i en .til brukeren, og da har det gått greit å snakke sammen.

E: Ja, men å komme fram til, la oss si, et overall design, et overordna design av systemet hvis dere var på en måte eksperter på hvert deres felt med lite overlapp av kunnskap.

K: Ja, det har ikke vært så veldig mye diskusjoner på det. Det har det ikke.

E: Kundens rolle, altså organisasjon A, da, sin rolle i det her, hadde de en like

aktiv rolle i utviklingsprosjektet, synes du, altså som dere hadde, altså, jeg mener, som bidro til ..?

K: Som brukere?

E: Ja.

K: De kom veldig sterkt inn etter hvert. Så ..Men til å begynne med, når vi sleit med designfasen, så hadde vi ikke de representantene fra organisasjon A som vi fikk etter hvert, da. Vi fikk inn brukere som visste hva de ville og ..problemet vårt var at det folka som vi hadde til å begynne med, til å designe systemet, de var bare med i designfasen, fra organisasjon A altså, så da de forsvant ut så kom de brukerne som skulle bruke dette her ..

E: De virkelige brukerne?

K: ..Ja, med en helt annen oppfatning, da

E: Ja, ja, nettopp.

K: Og da, da starta vi ..vi gikk veldig mange runder før vi kom fram til ..nå skulle vi liksom forsvare det som dem første folka fra organisasjon A hadde forklart til oss overfor dem som kom og skulle bruke systemet, og det.det ble liksom ..

E: Men hvem er det som hadde valgt .på en måte ..under prosjektetableringen lurer jeg nesten litt på, hvem er det som hadde valgt de personene fra organisasjon A som representanter inni .eller som deltagere i designbiten? Hvordan var det .

K: Det er organisasjon A sjøl, det er ikke noe vi har påvirket. Vi hadde bedt om å få folk, for vi veit at det er viktig å få folk.

E:Ja, nettopp.

K: Men vi vet ikke hvem altså dette er ikke noe folk altså, dette er ikke noen vi kjenner fra før. Vi visste ikke hvem vi skulle spørre etter.

E: Nei, nettopp. Så de blei bare allokert til prosjektet?

K: Ja.

E:Og så måtte dere bruke dem.

E: Ja. Nettopp. Og det ..hvordan vil du karakterisere det? Var det vellykka eller var det den første fasen, eller vil du si at det var problematisk?

K: Nei, vi oppfatta jo ikke den første fasen som problematisk kanskje ikke før vi kom over i den andre fasen av prosjektet, da de faktiske brukerne kom inn og .det var vel ikke problematisk på noen annen måte enn at det tok innmari mye tid, rett og slett. Med å ..med å skulle ja.

E: Mm

K: vi oppdaga jo at , liksom ..for at det blir veldig vanskelig når noe står i spesifikasjonen eh .og så er det ingen som forstår hvorfor det står der. Og så er det ingen som tør å ta det bort. Og det bruker vi veldig mye tid på. Men kanskje .ja, kanskje vi skal ha dette her likevel, og så er det liksom .sjøl om man kanskje ikke vet helt hva man skal med det.

E: Men den spesifikasjonen, kom den før den utviklinga av en prototypen? Eller var det et resultat av prototypen, eller var det .?

K: Ja, vi hadde jo ..jeg veit ikke, har H snakka om ..altså, vi hadde en sånn, du kan kalle det teknisk prototype eller arkitektur-release, da, som skulle være en teknisk prototyp, og få se at ASP og COM og så videre kommuniserte, og så ..men før det så hadde vi jo en sånn en som vi laga i stand med HTML-sider, om hvordan applikasjonen da .det var mer et forslag fra vår side, da, for det .på det tidspunktet i utviklinga, så var det vi som skreiv veldig mye av spesifikasjonen.

E: Mm

K: Og det ..så vi kom på en måte med et forslag på åssen dette skulle se ut,

og det det ..det kom jo ganske tidlig. Men så kom jo disse her nye og tunge brukerne inn som sagt, og .da gjorde vi om på veldig mye av det. Vi hadde liksom grunnkonseptet, men så ble det...

E: Ja.

A: Var det vanskelig for dere å forholde dere til det?

K: Altså, det .altså det tar tid. Altså, vi har hele tida opplevd det som om vi har hatt et greit forhold og prata godt sammen og sånn, men det .vi merka jo at tida gikk, da. Det var nok littegranne slitsomt for .ja. for de som kanskje, altså E, som sitter med brukergrensesnittet at det kanskje ikke var så avklart som han .altså som det burde ha vært. Og så kommer det endringer. Det er jo ikke akkurat noen ideell situasjon. Men jeg tror det at ..det føltes ineffektivt. Det virka sånn.

E: Føler du det at også jobbe på en måte sånn iterativt som du gjorde i begynnelsen særlig ...er det ineffektivt også? Er det det som egentlig er ineffektivt, altså at du stadig da får endringer på den måten? Litt mer generelt kanskje, hva er dine holdninger overfor det å jobbe iterativt, inkrementelt sånn som dere har jobbet?

K: Jo, altså ..da det virka ikke i dette prosjektet nå, og det blir veldig mye avhengig av bemanning altså, tidligere har jeg jobba mye mer etter kravspesifikasjon enn det vi gjorde her, altså vi fikk ..vi fikk veldig mye i henda som vi på en måte, vi tok jo på oss og prøvde å dokumentere noe, og etterpå så har jo organisasjon A angra på det, altså .de vil ikke ha det sånn igjen de heller. De følte at de, de mista liksom kontrollen med hva som skulle lages her, da. Som en sånn .en designprototyp det nei, jeg vet ikke helt, jeg, sånne generelle holdninger. Det avhenger da av systemets art og folka som er med og mye sånt, men det endte jo opp i en sånn pengediskusjon for vårt vedkommende i det prosjektet her nå. Vi kom ut med mye flere timer enn det vi hadde tenkt, da. Så kan man sikre seg mot det ..du må i hvert fall være klar over det at du vet jo på en måte ikke hva du skal lage før du er der, da. Etter du har holdt på en stund, og det gjorde det jo veldig vanskelig å estimere det her på forhånd. Så jeg vet ikke helt, jeg

A: Hva tror du de som har vært med først og designa .hva synes de om at det kom nye krav og ble et annet system

K: Nei, de de synes vel .altså, det er jo de som bemanna prosjektet sitt. Og jeg tror nok at sånn som det ble blei det nok det riktige, altså ..fordi at de første folka som var innom, det var folk med litt mere vyer og ...altså, de ville lage et system som tok høyde for mye. Men så kom det på en måte fotfolk og lagde et system for det de trenger her og nå. Og det er nok riktigst sånn som situasjonen er, for ting endrer seg fort, og så ville dem liksom ta høyde for masse framtidige ting som kanskje skal komme i framtida nå, og det gjør at det tar mye lenger tid å bli ferdige med det en trenger nå, og når vi blir ferdige blir det kanskje ikke sånn likevel i framtida, så ..vi tok kanskje litt for stor bit til å begynne med, at vi kanskje skulle delt det opp det er litt det samme som de gjør nå egentlig i fortsettelsen, vi ønsker å dele opp og ta litt steg for steg det vi vet skal gjøres. I stedet for, sånn som organisasjon A ønsker å ha veldig store moduler, da.

E: Mmm

K: Og du går litt i den fella at du aldri blir ferdig og du .det går på bekostning av det vi kunne bruke med en gang, altså, det man trenger her og nå. Da må man ofre litt for det som kanskje kommer i framtida så

A: Har du noe inntrykk av hva ledelsen synes om at dere har forandra det så mye? Hva de oppdragsgiverne synes om at dere har forandra det så mye? Var det greit for dem, eller .?

K: Ja ..altså, de har jo reagert på tids at vi har brukt så mye tid på det.  
A: Mmm  
K: Men .de har ikke kommentert i det hele tatt..altså, de er veldig fornøyde med det de har fått nå.  
A: Mm  
K: Det ..  
A: Men viste de forståelse for at der forandra det underveis, eller .at det var derfor dere brukte så mye tid?  
K: Nei. Det har vært et problem at de har . De har ikke sett det. Fordi de har ikke følt at de ja, atte, de har ikke sett hva som har skjedd og sånn, og så har de ikke har ikke gått med på det, liksom, at de har ..at det faktisk er de som er skyld i mye av tida.  
E: Ja. Har den kunden vært egentlig veldig på en måte bevist på rollen| til brukeren i kravspesifikasjonen, eller det virker på en måte som om de ikke har det, da i og med at de  
K: Ja, de brukeren var veldig lite med i den første kravspesifikasjonen og det var både vi og organisasjon A enige om at det var ikke bra.  
E: Nei. Men de vil ikke ta ansvaret for det egentlig?  
K: ikke da, nei Men nå, altså nå i neste fase så har jo de fått med brukerne sine mere sjøl, da. Da vil de ha en litt mer modell med en kravspesifikasjon som gjøres på forhånd, og vi leverer ..svarer med et leverandørtilbud.  
E: Mm  
K: på det som skal leveres, da. Med en fastpris, da, det ønsker de nå.  
E: Ja vel .Men da blir det ikke særlig iterativt og .  
K: Nei!  
E: Da blir det veldig fossefall her, hvor de utarbeider en kravspek som dere laver, og så aksepteres det eller ikke?  
K: Ja, de..ja, det er jo det som skjer nå. Så de ønsker ikke å gjøre det for de, de følte de fikk for liten styring på prosjektet, rett og slett, da. I og med at de hadde noen foreløpige estimater som de la fram for sine investeringsforum og så blei det ikke sånn, så blei det mye dyrere og de vil ikke ..de kan ikke .som kunde, gjøre det en gang til.  
E: Mm. Nettopp.  
K: Så det ..  
E: Hvem er det som skal utarbeide kravspesifikasjonen for dem nå da? Er det brukerne nå da, forhåpentligvis?  
K: Ja, de har gjort det sjøl nå, ja. Og så har vi kommet med tilbud da, tilbake. De har liksom tatt littegranne rom for endringer, men ikke den ..ja, vi har også samarbeida med organisasjon A i å skrive kravspesifikasjonen, da, men det er de som har ført pennen.  
E: Så utifra det du sier nå, disse erfaringene her, kan du på en måte generalisere noe utover dette prosjektet her? På den måten, at altså, virker det slikt at Det inntrykket jeg har nå, etter å ha snakket med to av dere, er at kanskje den type evolusjonær iterativ inkrementell Rational Unified Process – prosesser den kanskje egner seg veldig godt for mindre interne prosjekter, men kanskje ikke så godt når du har en ekstern når du er en ekstern leverandør til en kunde? Så det virker nesten sånn at det er jo de som nærmest her vil ha litt mer sånn formaliteter og litt mer sånn faser og leveranser på den gamle gode måten, for de får da mer kontroll. Stemmer det, er det riktig analyse, for det første, og kan ... hvis det er

riktig, synes du at du kan generalisere det utover det prosjektet her?

K: Jeg tenker på ..jeg tror ikke jeg har vært med på så mange typer prosjekter før, jeg, som har gjort det på den måte, altså, det er mye mer vanlig her hos oss i hvert fall med enda mer kravspesifikasjoner i en eller annen form. Dette vi vi ha løst, og .ja, sånn at jeg vet ikke om jeg kan si så mye mer generelt. Altså, det var ikke noen suksess for oss nå. Men om det da .det kunne vært annerledes på en annen bemanning, altså. Det men det vet vi jo ikke når vi starter opp.

E: Nei .

K: Sånn at, eh ..

A: Hvis du skulle valgt på nytt igjen, ville du valgt en annerledes prosessmodell da?

K: Altså, hvis vi skulle starta opp et helt nytt et tilsvarende prosjekt med helt ny kunde som ?

A: Mm

K: Nei..jeg vet ikke. Det er nok det komme litt an på, altså. For det er veldig sånn ..altså, det er jo hva kunden ønsker også da. Det er jo kunden som gjorde at det blei sånn som det blei nå sist. De ville ha med oss på å skrive den kravspesifikasjonen eller den ..designe det systemet, da, med skjermbilder og sånne ting. Og så meldte de seg litt ut sjøl og. Jeg vet ikke om vi ..om vi kunne gjort noe kunne stoppa det. Men det er vanskelig å si, for du sitter på en måte oppi det, du oppdager sånne ting litt seint.

E: Men det er vel en annen ting, at det er en litt spesiell situasjon her at kunden her er jo også veldig teknisk kompetente, går jeg ut i fra.

K: Ja, organisasjon A er en veldig stor organisasjon. Så men folka som vi hadde i vårt prosjekt var ikke det.

E: Nei, vel.

K: men tror du ..tror du de er i stand til å skrive dem kravspesifikasjonen på en ordentlig måte?

K: Sånn som han som jeg nevnte der, blant annet, han har kommet inn etterpå. Han var mer analytisk. Skriver litt .tenker litt mer i system, ikke bare ..funksjoner og Jeg husker ikke hva du spurte om igjen. Ja, du starta med å spørre om ikke de hadde kompetente folk.

E: Jaja. Det er bare mer sånn, altså ..jeg prøver å tenke litt på .her ser det ut til at det heller mer mot sånn .litt mer sånn fossefall, men altså, forutsetningen for at det også skal fungere må jo være at kunden virkelig er i stand til å ..at for det første at de vet hva de skal ha, og at de er i stand til å beskrive det på et detaljeringsnivå som gjør at dere kan egentlig notere det da

K: Mm. Det er ikke sikkert at det nødvendigvis var metoden som ble det største problemet for oss, men nettopp det at det ble det byttet av bemanninga. At du hadde noen folk med vyer i analysefasen, og så var det ut med de og så kom de andre inn. Jeg tror kanskje ..om du hadde brukt den ene eller den andre metoden så, uansett. .det er ikke sikkert det hadde så mye å si.

E: Nei. Så at kanskje et råd, for å si det sånn, er som sagt, det her er jo en erfaringsuthenting for å prøve å lage en bedre prosess. Så et innspill fra din side høres ut til å kanskje være at når man etablerer brukerkontakten, så sørge at de er der fra svært tidlig i prosjektet, og at de ..det kanskje er de samme menneskene i en lengre periode, da?

K: Ja, det er nok litt smart, ja.

E: Det er ganske riktig?

K: Mmm

E: Nettopp. For det er også litt interessant at det her er det andre webprosjektet jeg har vært på som .enten har intervjuet eller vært med og observert, hvor jeg ser noe av det samme, at kunden ser litt lett på det å spesifisere hva en bruker av en sånn type webapplikasjon egentlig er. Kanskje ikke spesifikt for webapplikasjonen, men å få en representativ bruker, og å ta det ansvaret litt alvorlig, det høres ut som om det har gått kanskje litt galt her? Jeg vet ikke .at det kunne vært gjort bedre, da.

K: Ja, jeg tror nok det at dem undervurderer den tida det tar med alle småtinga, da, som ..og særlig når noe er bestemt på forhånd som man skal gjøre om på, så tar det enda lenger tid. Men vi var for så vidt klar .vi etterlyste brukere helt ifra begynnelsen. Det tok litt tid før de fikk etablert det, så det var ..Men de må jo få frigjort de folka de skal ha også. organisasjon A også, de også var klar over det, de også jobba med å å de folka de trengte.

E: Men brukere er en ting, en annen ting .jeg tenkte på, sånn testfasiliteter, testressurser, hvordan var den arbeidsfordelingen der mellom kunden og dere, da? Hadde de .leverte dere del-leveranser som ble levert, og installert og på en måte tatt i bruk, a la en pilot?

K: vi hadde ikke del-leveranser i den forstand, men vi hadde jo alt tilgjengelig hele tida på ..eh..ja, de kunne jo sitte og se hvor langt vi hadde kommet hele tida. Vi hadde det jo ut på server og de hadde tilgang til serveren og fikk logga seg på. Så det så jo .de så på at vi jobba, for å si det sånn. Men .men det var .det var vel sånn at .vi har vært veldig fornøyd med de brukerne vi har hatt, da. Når de først kom så har de testa veldig mye. Hjelpet oss veldig mye i test og vært med ..

E: Men det høres ut som om det har fungert veldig bra, da, at dere har fått et system opp å kjøre egentlig som .ja, ikke sånn i drift, men sånn at ..i hvert fall sånn for uttesting ganske tidlig. Er det riktig?

K: Ja, siden det er en webapplikasjon, så har vi .vi har fikk etablert miljøene våre med testutviklings- og produksjonsmiljø, og vi har veldig godt opplegg på .vi har en testansvarlig her som hadde jobben med å koordinere testen og skrive testscript sammen med brukerne, så .vi har også et verktøy som heter testdirector om brukes til å til å .legge inn testscript og kjøre da testscripta fra testdirector og kvittere på om det var OK eller ikke og .så det er veldig sånn godt organisert testmiljø her. Det sammen med de brukerne som var veldig ivrige, da, så fungerte det helt utmerket.

E: Dere har hatt .dere har hatt forskjellige servere selvfølgelig. Utviklingsmiljøet deres var jo separat fra

det de testa på, går jeg utifra? K: Mmm. Det er tre fysiske miljøer som vi har

E: Nettopp. Men hvor ofte porta dere .tok utviklingsmiljøet og porta over til den som de testa på, liksom?

K: Nei, det ja, først da, så hadde jo de bare tilgang til vårt utviklingsmiljø, da. Mens vi holdt på. Siden så blei det .ja, så var det jo feilretting under testen, med mer ujevne mellomrom, altså

E: Men i begynnelsen så hadde de bare tilgang til .direkte til utviklingsmiljøet?

K: Ja. Vi slapp de inn der, og .

E: Ja, vel .og det var greit, liksom?

K: Ja, for oss eller for de?

E: Ja, begge deler, kanskje?

K: det er jo ikke så mye glede de får av det når de sitter og .men de var jo inne og

kikka litt, da .Vi så ikke noe ulemper ved det. Det var liksom ikke noe hemmelig ..

E: Nei. De var selvfølgelig da veldig klar over at det var utviklingsmiljø, da, at det ikke var .

K: Ja, altså så lenge de gidder å gå inn å se .altså, for oss var det ikke noe ..tror ikke vi tapte på det. Det er klart, du får litt spørsmål og sånt, og kommentarer .men vi gjorde det sånn, da.

E: Interessant. Ja, jeg Vi har jo vært inne på ganske mye nå med kundekontakt og ..Kan du ..kan du si noen om hvem som på en måte har vært initiativtakerne, eller premissleggerne eller stakeholders eller hva du nå kaller det på prosjektet. Hvem er det som .altså, det ser jo ut til at du har forskjellige typer brukere inne. Ekspertbrukere i begynnelsen som hadde vyer om hvordan dette skulle gjøres. Og hadde de “ordentlige” brukerne etterpå som ville ha et litt annet system. Jeg lurte på, hvem var det som på en måte egentlig ville ha det her systemet i begynnelsen. Var det .var det noen som ville ..skjønner du hva jeg mener?

K: Ja. For jeg tenker på prosjektledelsen, hvorfor stoppa ikke de de med vyer hvis det var sånn at det var .

E: Hvem var det som egentlig initierte prosjektet, altså, hva slags personer var det som følte at de hadde mest å tjene på som var mest interessert i at prosjektet ble gjennomført?

K: Hvem det er, altså, det er helt konkret montører, altså de som jobber med kabelomleggingen .det er de som skal bruke dette systemet.

E: Men var det de som ville ha det?

K: Nei, altså, det er nok de som ville ha det, men så ble jo det da ..en enhet i organisasjon A som tar ansvar for å kjøre prosjektet...vil da dra inn brukere av...altså, de folka som ble med først, de er jo ikke brukere som sagt, da. Det er jo ikke de som skal bruke det, men de vil jo også ha det, så jeg vet ikke helt hva jeg skal si .men de burde kanskje sett det sjøl. Burde kanskje få opp noe først, og så heller bygge på litt seinere, i stedet for at det ..

E: Jeg prøver å finne ut av nå har man nærmest bestemt seg for at, OK, nå tar vi det litt av gangen av det brukerne vil, men i begynnelsen så virka det som om det ikke nødvendigvis var det. Altså, derfor lurte jeg egentlig litt på da, hvem var det på en måte som ville ha dette prosjektet?

K: Vi kunne jo ikke se det. Vi visste jo ikke noe om det før vi begynte med det selv.

E: Nei.

K: Prosjektledelsen er også litt i samme situasjon som oss, at de hadde heller ikke noen detaljspesifikasjon på dette her. De hadde bare fått ansvar for å kjøre prosjektet. Men nå er vi litt tilbake på den byttinga av hovedpersonene på spesifiseringa. At de hadde litt forskjellige oppfatninger .personavhengige.

E: Tekniske ting ..Altså, dere designa en del av systemet i UML har jeg forstått? Var du med på den biten?

K: Nei, det var stort sett K og C som ..de starta jo opp og modellerte metoder og .men .jeg føler at jeg har fått veldig lite ut av UML egentlig, altså.

E: JA. Og Rose-verktøyet?

K: Ja, på grunn av noen tekniske problemer i forhold til Rose og at vi bruker ATL og .Og støtten fra Rose til ALT som de hadde problemer med når de brukte Reverse Engineering fra C++-kode til Rose så fikk de ikke med kommentarene, og det er så mye manuell jobb. Pluss at det var en høy terskel, rett og slett, med å lese sekvensdiagrammer og Use Cases, du har aldri fått brukeren inn i Altså, vi



har ikke hatt noe nytte av dette i forhold til brukeren.

E: Nei, de har vært mer interessert i litt høyere nivåspesifikasjoner, eller skjermbil-despekk .

K: Ja, litt sånne ting, og kanskje det gammeldagse flytdiagrammet. Men Rose fikk vel det i den nye versjonen, men så var det ikke støtte for det til å begynne med.

E: Nei.

K: det var ikke den store suksessen med sånn som det er nå, så er ikke UML-modellen vår oppdatert i forhold til sånn som applikasjonen er. Det har glidd litt ut ..

E: Ja, det høres jo ut som om det er en erfaring dere deler med veldig mange, da.

K: Det virker som om det er mer en heft enn en hjelp, liksom. Du skal ha den dokumentasjonen i tillegg til at du skal lage systemet, så er det klart at

E: Og så har jeg ennå ikke møtt på, etter å ha vært med på mange utviklings-prosjekter, har aldri sett noen som virkelig har holdt designdokumentasjonen "up to date" med kode.

K: Nei.

E: Det ser ut til å være nesten en umulig oppgave. (uklart) Sekvensdiagrammet og UseCasen må du oppdatere manuelt, så .

K: Hva kan vi gjøre med det?

E: Nei, det er det som opptar mange nå Vi bare konstaterer det foreløpig. (litt om forskning i ISU, .)

K: Nei, jeg synes ikke vi har hatt så stor nytte av det (UML/Rational). Det eneste positive med det var at K og C satte opp sekvensdiagrammer og brukte det for å identifisere metodene som skulle lages og sånne ting, så det var jo ikke så dumt da. .men ellers sånn, så er det det med vedlikehold, som du sier..det har ikke blitt vedlikeholdt.

E: Men for din del, sånn fra databasesiden, det var du som tok den Oracle-biten. Du synes ikke du hadde noe bruk for ..

K: Vi har jo hatt et annet sånn verktøy som heter Data-arkitekt, noen gamle greier som egentlig er helt kurante. Det brukes som datamodelleringsverktøy, da, til å lage både fysiske og logiske modeller, og generer databasetabeller og sånn fint. Litt indekser og sånne forskjellige ting.

E: Det gjør det, ja.

K: Vi har blitt anbefalt, egentlig, å ikke bruke Rational Rose i databasemodeller-inga mot Oracle fordi at det var ikke ..eller jeg prøvde det ikke sjøl, da, men det var visst mye...ja

E: Ja, så den fancy OO-verktøystøtten den har ikke vært ..akkurat det aspektet har kanskje ikke vært så veldig vellykka, egentlig ?

K: Nei, det har nok ikke det altså. Det ..B har jo vært mere aktiv i detta, så han kan jo svare litt mer ..

E: Mm

K: Nei, for jeg har prøvd å legge litt Use Case-modeller inn i dokumentasjonen og sånt, men det er liksom ikke noe som når så veldig det blir for grovt. Eller så blir det .det er ikke noe spesielt opplagt, de .på en måte så er det kjempeenkelt, men på en annen måte så gir det meg ikke noe.

E: Nei. Men jeg lurer på ..littegrann..hvordan er det at det her ..altså, de har lag et opprinnelig design for da, hvis jeg skjønner riktig, på en måte det mellomlaget, med MTS og de komponentene der. For COM og ATL og C++-klassene og alt det her. Det er generelt det dere har lagd UML-klassermodell for?

K: Ja. Det stemmer.

E: så tenkte jeg at det burde vel egentlig få noen konsekvenser inn mot datamodellen, burde det ikke det? Eller gjorde det det? Eller hvordan var det?

K: Jo, det er klart, det henger jo sammen. Du kan si at siden de at parallelt med at de jobba med det, så jobba jeg med datamodellen.

E: (Uklart, litt prat i munnen på hverandre) Men det er liksom to likemenn som på en måte prøver å bestemme designet, er det ikke det?

K: Njooooo, jeg oppfatta ikke det som noen konflikter. Kanskje egentlig jeg var først .At de så på hvordan databasen var, at det ble mer sånn.

E: Så styrte de designet utifra datamodellen?

K: Ja. Egentlig det. Vi har ikke gjort mye om på databasen, den har ..

E: Nei, OK.

K: Så det ble jo litt sånn derre ..ja altså, den er jo laga uten noen sånn spesiell teknikk, jeg bare visste jo funksjonaliteten , og designa utifra det som skal leveres. Det blir på en måte en parallell teknikk, bare at det blir ikke noe ..det blir bare utifra erfaringer.

E: Ja. Det er jo for så vidt også interessant, når du tenker på ..Det er jo andre komponenter, la oss si at det er andre komponenter der også, så er det jo typisk .når det gjelder personer med forskjellig erfaring og med forskjellig bakgrunn vil typisk ønske å bruke sine spesifikke verktøy. Det virker jo på meg som om det er noe med det metodiske eller prosess som egentlig .du sier at i det tilfellet her så blei designet da egentlig, i bunn og grunn avgjort av datamodellen. I veldig stor grad.

K: Ja, når det gjelder ja, de tinga du sa i stad som har med data å gjøre, kan du si, så ..da var det nok den som styrte.

E: Ja, nettopp. Hvis du tenker tradisjonelt, Jacobson eller disse guruene innenfor objektorientering, innen objektorientert analyse og design, så tenker de jo seg alltid at du har en objektmodell, og utifra objektmodellen skal det spinne ut en datamodell .

K: Det var ikke sånn det skjedde her, i hvert fall. Det var ikke det. Sånn sett så burde jo datamodellen komme til slutt, og være et resultat av ..

E: Ja, hvor alle da deltok i den design

K: Ja.

E: biten.

K: Sånn sett er det kanskje rart at det det gikk jo veldig smertefritt, da. Du kan kanskje lure på hvorfor det, egentlig. På hvorfor det ikke var flere konflikter, for det har det ikke blitt, altså, sånn mellom komponenter, metoder og

A: Jeg lurte på hvis du skulle gjort noe annerledes i prosjektet, hva ville du gjort annerledes da?

K: Mm. Nå har vi jo snakka om det med brukersida, men så har du jo det tekniske. Vi gikk jo løs på dette her sånn ..det var ingen som kunne noe om det, altså, vi .det var litt sånn hodeløst valg av plattform, synes jeg da, at en skulle bruke COM som mellomvare, og så ble det COM fordi det var noen andre her på huset som hadde noen COM-komponenter som vi kunne få, og så skulle vi liksom gjenbruke de litt, og så ..så kom K, da, med MTS og ..ATL og så bare blei det sånn. Og det var et veldig .sånn i ettertid vi ville ikke og så ble det C++, da: ja, hva skal vi velge? Jo, vi tar C++, liksom

E: Ja, vi har kompilatoren oppe, liksom?

K: Ja, fordi vi snakka med en her: ja, det er fint med C++, det er fint, det er

mye raskere enn vi snakka litt om Java og sånn. Hvis vi skulle valgt plattform nå, så ville vi nok valgt Visual Basic. Og vi ville nok sikkert også, når det gjelder kommunikasjon av databaser, så bruker vi Oracle siden (mumling om tekniske komponenter). Men i hvert fall, det virker i hvert fall ikke med MTS, da, men vi har hørt av andre som kjører med Oracle og ADO, som liksom er Microsoft sin databasekomponent, virker heller ikke med MTS, så det Men sånne ting, det tror jeg .Vi faktisk ikke fått konfigurasjonsstyringa vår til å virke enda ..Vi kjører eller transaksjonsstyring, mener jeg ..så vi kjører jo tildels uten. Og så at vi får oppdatert noen databasetabeller her og der som ikke skulle vært oppdatert. Det er jo ikke noe kritisk heller, men det er klart vi har ikke fått til verken vi hadde for dårlige greie på, altså, det burde vært veldig..vi sliter jo litt firmaer burde få en litt mer sånn ensarta politikk på hva de velger av verktøy rett og slett, altså. Det burde ikke vært opp til hvert enkelt prosjekt å bare velge noe, sånn som vi har gjort .med den kompetansen som vi har.

A: Mm

K: I hvert fall sånn som da K kom hit og jobba med oss og ..Vi hadde nok spart en del tid på å valgt Visual Basic, da, som Vi har jo ikke noen sånn spesielt komplisert forretningslogikk som skulle tilsi at vi ..måtte bruke C++. Vi visste ikke da hvor mye det kompliserte Det, og at vi ..på det, kall det arkitektur-release da, den tekniske prototypen, så tok vi litt for mye inn i den. Vi brukte for mye på den. Så viste det seg at når vi var ferdige med den, så var det faktisk så mye TelMont-applikasjoner som lå i det at vi kunne ikke ombestemme oss. Selv om vi kanskje kunne hatt lyst til å gjort det, da. Så da blei et sånn.

E: Det tok ganske mye tid?

K: Det tok mye tid, ja, og vi la nok litt for mye funksjonalitet i de.

E: Så da ble den for tung å kaste?

K: Ja, den gjorde det.

E: Så den prototypen blei systemet?

K: Ja. For når vi var ferdige med den, da var vi vel blitt littegrann sånn klar over det her med at vi kanskje heller burde ha satsa på Visual Basic. Det er kanskje det jeg helst ville bytta ut hvis jeg skulle begynt igjen på nytt, da.

E: Men det er sånn veldig tekniske ting, da, som du først og fremst ser at kunne vært gjort annerledes? K: Ja, men det er noe med at vi som tok den avgjørelsen..vi som var i det prosjektet, ingen hadde brukt den teknologien før i det hele tatt, og da blir det feil at det er vi som skal ta den avgjørelsen også, at det er vi som skal ta det valget. At man skal ha litt mer styring på verktøyvalg, rett og slett, altså. Og plattformvalg. Men, vi var jo innom alt mulig rart. Unix og ..alt var helt åpent, da.

E & A: Mmm.

K: Nett hadde ikke noen føringer i det hele tatt. Vi visste at det skulle være en webapplikasjon, da

A: Men et annet spørsmål: la dere veldig mye vekt på å gjøre applikasjonen konsistent layoutmessig?

K: Nja, det Nei, egentlig, så ..Hvis du ser på den nå, så er den jo faktisk ikke det. Så det er ikke sånn for eksempel .hvis du ser på feilmeldinger i bildet for eksempel, så ser ikke de like ut. Noen ganger kommer det pop-up opp og noen ganger kommer det vindu og .det er litt sånn .så, vi kunne nok tenkt litt mer på design, ja, altså på brukergrensesnitt. Men .det er ikke noe katastrofe, heller, men du ser at det er ikke ensarta.

A: Nei. Nei.

K: K: Ikke hundre prosent, da. Det er jo ikke at alle bildene har forskjellige farger og sånn, men de er sånn noenlunde samme, men blant annet det med feilmeldinger og litt sånn hvor kursoren styrer seg og .når du kommer inn .altså, hvor du har tilbakeknapp og litt sånn forskjellig. Det er litt sånn tilfeldig. Ikke fullt så ordentlig

A: Har du vært med på å lage et sånt navigasjonskart over hva som skal skje når du trykker på den og den linken og sånn?

K: Vi har jo hatt sånne ..vi har ikke akkurat laga noe sånn navigasjonskart, men vi har .vi har hatt flytskjema, da. Som viser en sånn normalprosess. Det har vel Nett skrevet en del på sjøl. Så det er det vi har av den .

A: Var det nyttig for dere, på en måte?

K: eh vi brukte så mye ..Altså, vi har snakka veldig mye om det. Vi har ikke skrivi .Men det flytskjemaet har absolutt vært nyttig, altså. Åssen de tenker seg prosessen .

A: Ja.

K: det er nyttig for oss. Og for alle parter. Det er liksom noe alle klarer å følge, liksom.

E: Det er kanskje bedre enn sånne Use Case-modeller?

K: Jeg synes egentlig det. Du ser det er jo det, når de skal lage noe sjøl, kunden, så er det jo det de ..det de lager, og det klarer jo alle og se at: ja, her går det liksom..du går ut her og inn igjen her og ..

E: Men hvordan har dere gjort det sånn helt konkret? Teknisk, hvordan ser et sånt hvordan har det sett ut, liksom, det som brukerne sånn intuitivt har tegna opp, på en måte?

K: Vi har tegna det i Word, altså, helt enkelt. Med firkanter og bokser og sånn ja, valgbokser og operasjoner og sånn, da, at de har symboler for det, da. Men det er klart, vi har jo tegna flytskjema på noen standardoperasjoner, men du har jo andre muligheter, du kan jo gå veier i applikasjonen som ikke er dokumentert sånn da ..gjøre ting i annen rekkefølge, så de flytskjemaene er for de standard prosessene, den standard arbeidsflyten ..

A: Måtte dere endre de flytskjemaene underveis i prosjektet?

K: Vi endra de litt også. Jeg husker ikke akkurat sånn ..

A: Nei.

K: Men det har ikke vært det som har vært mest ..Det har vært endringer som har dratt med seg endringer i de, liksom.

E: Men når de virkelige brukerne begynte å komme med sin innputt, så var det altså ..Hva var det hvis .det var ikke så mye flyten? I applikasjonen, eller var det ..Altså, hvis du skal prøve å karakterisere, da, hvilke aspekter ved applikasjonen gjorde .endra de mest? Var det flyten eller var det databaseskjemaet eller var det .

K: Nei, det var nok det at vi ..i utgangspunktet så skulle alt være så veldig fleksibelt, men egentlig så ønska de mer en styrt prosess. Så sånn at applikasjonen vår egentlig var at du kan gjøre operasjon a, b, c, d helt uavhengig av hverandre, og med all mulig slags innputt og sånn man kan gjøre alt mulig med. Men det var ikke det de ønska egentlig, de ønska .det de skulle var å gjøre a, b, c, d i den rekkefølgen og ikke ha så mange muligheter. De ville bli litt styrt og at applikasjonen viste deg hva som var det neste steget, da, så de ..ja, de ønska en enklere applikasjon. Rett og slett. Derfor var det mer en støtte for den prosessen vi skulle gjøre, mer enn et verktøy for alt mulig rart, altså å sette det sammen .

A: Var brukerne med på å utforme det brukergrensesnittet deres?

K: Eh..ja, det var veldig mye styrt av altså, vi stod jo vi tegna jo HTML-bilder og litt sånn, og la fram for de, og da blei basisen endra litt på farger og men de hadde ikke så sterke meninger om det. Det er ikke sånne jåler altså, de skal bare ha noe som virker, de ..

A: Nei.

K: Noe de kan gjøre noe med.

A: Men viste dere de til dem og ?

K: Ja, vi viste dem til dem ..Det er veldig nyttig det også da la vi dem jo ut på serveren og laga noen linker mellom linker og sånn. ..De fikk jo inntrykk av applikasjonen ut fra det, da. Men de er veldig sånn nøkterne..Ja, ikke noe sånne snobber som skal ha masse greier som det går an de har stilt veldig lite krav. Det finnes kunder som er mye verre enn disse her når det gjelder ..ja, masse småpirk da, som de ikke brydde seg noe om.

E: Det virker som om ..Det er veldig mye nyttig informasjon her for oss ..Absolutt.

K: Ja, for vi gjorde feil det var to ting: det med spesifikasjonen vår, hva vi egentlig skulle gjøre i forhold til hva vi starta med, og så var det det med teknologivalget, synes jeg. At vi gjorde det på den måten som vi gjorde det, uten vi skulle kjørt en teknisk prototyp (måtte snu kassetten.) Men på prosjektet ellers, var det et godt forhold mellom brukerne og oss, da.

E & A: Mmm.

K: De har vært veldig ivrige på dette her de da, de har jo vært her en del ganger. De kom fra Oslo forskjellige kanter av landet og har sitti her og testa og ..

E: Ja, de har det , ja?

K: Ja, så det er litt sånn spesielt, tror jeg, at kunden har reist mye. Jeg har nesten ikke reist i det hele tatt, jeg. Fordi ..De kom fra Bergen og Gjøvik og Oslo.

E: Så de er interessert i å få

K: JA, det er de, veldig interessert i å jobbe med det, å bruke tid på det. Og godt fornøyde med det de har fått, da.

E: Ja. Det var bare kanskje litt sånn feilskjær i begynnelsen egentlig?

K: Ja, det var å komme ordentlig på sporet funksjonelt. Og også teknisk egentlig. Men hele tida har det vært .veldig ålreit forhold, synes jeg.

E: Ja.

K: men hadde vi ikke hatt de folka fra Nett så mye, så .kunne vi blitt, (latter), ja da hadde det blitt noe helt annet.

E: Ja.

A: Har du noen formening om hva du mener er forskjell på å utvikle web-applikasjoner og andre typer systemer?

K: Nei, fordelen sånn som den applikasjonen vår er nå, er at .nå ønsker de å få dette her ut på Internett, blant annet. Jobben det systemet her gjør, er at montørene til organisasjon A som står i sentralen og kobler telefonlinjer, de vil gjerne kvittere ut at de har gjort det fra der de står, da. Og da er de jo ikke inne på det lokale nettet vårt, og da skal de ha tilgang ut på Internett, da. Og, det er klart, i og med at vi har en sånn arkitektur fra før, så burde det i hvert fall være enklere å få til det enn om vi hadde valgt de...Powerbuilder eller Visual Basic eller et eller annet.

A: Mm

K: det er bare sånne sikkerhetsmessige ting som stopper nå. Så er det jo .jeg har jo holdt på med Powerbuilder før og jobba på stormaskin og ..det er jo klart, det er jo en fordel installasjonsmessig, at du slipper å (uklart) klienten.

A: Mmm. Men selve prosessen, er den ganske lik?

K: Ja, du har denne den fordelen som jeg sa nå, og så installerer du ..du må jo installere ny versjon på serveren i alle fall.

A: Mmm

K: Og ..Så det er jo ..du får jo en god lagdeling på den måten, liksom. Det gjør du.

A: Mmm

K: Men..brukergrensesnittet sånn sett i i ..Internett explorer og sånn, det er jo begrensa i forhold til hva du kan gjøre med andre verktøy, sånn som for eksempel Visual Basic, da

A: Ja

K: Sjøl om det blir bedre, og det blir sånn browseruavhengighet og sånn, det er jo ikke akkurat Vi har jo bare laga for Internett explorer vi nå, da, men ..Det er jo ikke sånn at det blir browseruavhengig sånn av seg sjøl, liksom, så da, at det er ..

E: Nei, det gjør det ikke.

A: Har dere lagt mye vekt på sikkerhet og sånn?

K: Ja, det...Det er jo en av de store sakene som...særlig nå i forbindelse med at de skal ha .kunne koble seg opp mot applikasjonen utenifra, så har en del sikkerhetsaspekter vært veldig mye snakka om, da, men .Det er vel noe snakk om at organisasjon A holder på å legge opp en sånn mellomvarepakke. En plattform ..Flex-it og Flex-it metro, to forskjellige sånne .som også har et sånn sikkerhets .så sikkerheten på applikasjonen nå er egentlig ganske enkel, det er sånn ..eh .har et sånt rammeverk som er laga her på huset som heter TWD, organisasjon A web desktop, som har en liten database i seg som du legger inn brukere, og som du får tilgang til en del applikasjoner som alle går under den TWD. Så det er veldig enkelt sånn .det er ikke noe sånn krav til passordbytte og .jeg vet ikke om passordet ligger kryptert på den databasen eller, jeg er ikke helt det er ikke noe sånn å skryte av det.

E: Den ligger utenom den applikasjonen, på en måte?

K: Ja. Det gjør den. Det var noe som fantes der før som brukes på noen andre webapplikasjoner.

E: Du trenger ikke gå i detalj på det tekniske, det høres jo vanskelig .Altså, hvordan Du blir autentisert på ett system, på en måte?

K: På en server, ja. Så da blir du utifra det systemet du logger på, da, så blir du sendt videre i systemet. Så fordelen med TWD, da, det er at du har ..da er du logga på en gang for alle. Så atte da alle systemene som du har tilgang på via TWD, de har du da som bruker.

E: Du får en nøkkel nærmest, da? Som du da bruker videre i de applikasjonene som sier at jeg har blitt logga på i TWD?

K: Ja, og da ser TWD hvilken applikasjon du får lov å logge på, og da vet den .da er det noe sånn securitylevel på det, da, som du tar med på den applikasjonen, sånn at applikasjonen, TelMont da, kan se hva du har lov å gjøre. Det er det som TWD vet om. Det er hvilken applikasjon du har tilgang til, og på hvilket nivå da. Så nivå 2, for eksempel, kommer du til TelMont med det, så blir du bare logga ut. Det er enkelt, men det er noe som .som brukes her på huset, da. Så sikkerhets det er liksom ikke ..det er ikke noe sånn derre ..Det har vært mye snakk om engangspassord for pålogging og så er det kryptering og saker og ting. Men så er det ikke noe. Men samtidig så er vi jo bare inne på internettet, da, så det skal jo ikke være ..

A: Mmm

K: andre enn organisasjon A-folk som har mulighet til å hacke seg inn hvis det

skulle

A: Nei.

(litt om hva mer vi skal spørre om)

E: Det er en ting, forresten, det er dette rundt feil, da. Har dere hatt mye feil fra spesifikasjon, design, kode, har det vært noen deler, på en måte, hvor du synes det har vært mest ..feil der?

K: Mmm. Vårt system skulle jo gå opp mot andre system, som sagt, så det største problemet har vel kanskje vært i forbindelse med altså, tenker du på under test eller i funksjon nå? For i test så hadde vi jo feil overalt, liksom ..

E: Ja.

K: Men nå som vi har vært i produksjon så har det vel egentlig stort sett vært feil i andre systemer.

E: I grensesnittet og sånn?

K: Nei, ikke i grensesnittet egentlig, men de funksjonene som de har lagd hos seg, da.

E: Ja vel.

K: Så

E: Ja, du mener mot det derre AXE og ..

K: Ja, det går jo gjennom et system sentralt altså, oppdateringa fra TelMont går gjennom et system som heter (uklart) og det er det som gjør .som foretar kommunikasjon igjen mot sentralen som gjør oppdateringer hos seg sjøøl og sånn. Så at det er feil der, da. Så sjøl om interfacet er ..så selv om vi sender en transaksjon dit og får svar tilbake og sånn, så er det kanskje ikke riktig allikevel.

E: Hva med underveis i utviklingen, da? Før det blei, liksom, satt i drift?

K: Nja, skal vi se det var mye problemer med de eksterne systemene da og, og også mye .vi fikk det jo så travelt, så vi dro jo inn K og så han S, (C++programmerere) Vi dro jo inn masse folk bare for å få koda ..De var jo her en stund, og så dro K og så dro S etter å ha programmert en stund, og det blei det mye feil av, da. De hadde jo ikke fått testa før de dro, ikke sant. Jeg veit ikke hvor mye vi fikk ut av S, sjøl om han var mye med og sånn, men han var akkurat så mye med at han akkurat fikk gjort noe og knapt nok enhetstesta det før han dro ut igjen, så er det det er ikke noe særlig lurt, da. Først skulle du sette han inn i noe, og så skulle du finne ut hva han hadde gjort etterpå. Så det så det var mye feil overalt, egentlig. I testen vår til å begynne med altså, da hadde vi mye feil. Men vi starta nok å teste litt tidlig sånn sett. På grunn av at vi hadde det travelt. Det er jo også en vanlig ting.

E: Ja. Du synes dere begynte å teste tidlig, ja?

K: Altså i forhold til altså, vi måtte jo begynne da. Vi hadde jo hatt utsettelse flere ganger, så .

E: Ja.

K: Men vi hadde nok med fordel kunne hatt systemtesta det litt mer sjøl i forhold til det .før vi fikk en offisiell systemakseptansetest, da.

E: Men hvorfor det?

K: Nei, for å rydde unna littegranne mer feil, så vi slipper brukeren løs på det. Det er klart det .

E: Ja, du mente det var litt for mye overhead, liksom, i forhold til det

K: Ja, for da har du plutselig en ordentlig .Vi kjørte systemtest og akseptansetest samtidig og da skal jo .da skal vi det blir mer folk som sitter og venter på når det stopper opp ting.

E: Mmm.

K: Det blir mer overhead av det.

E: Ja. Men dere har også levert flere systemer, har dere ikke det. Det er ikke bare en gang dere har levert fra systemet til noen, det er ..

K: Faktisk så er det det altså.

E: Ja vel.

K: Vi har ..vi leverte det før sommerferien, men det blei ikke tatt ordentlig i bruk før nå etterpå, men .vi har hatt bare en leveranse. Så har vi begynt på, eller gjort en del på neste.

E: Mmm

(runder av) Kassetten stoppes.



## Tillegg D

# Spørsmål til e-postintervjuene med brukerne på TelMont

Dette og det neste vedlegget inneholder spørsmål og svar fra de fire e-postintervjuene med brukerne fra TelMont. De eneste endringene som er gjort i svarene, er anonymisering av organisasjonsnavn. Spesielt er kundeorganisasjon kalt “organisasjon A”, mens utviklerorganisasjon er kalt “organisasjon B”.

### D.1 Intervjuspørsmål:

1. Hvis du kom med forslag til endringer av produktet underveis i utviklingen – hvor godt følte du at disse endringene ble mottatt?
2. Ble endringene utført?
3. Ble du fortalt at disse endringene kostet penger, eller fikk du inntrykk av at det ikke var så dyrt å innføre endringene?
4. Følte du at endringsforslagene du kom med var nyttige for at produktet ble mer tilpasset deg som bruker?
5. Likte du å være med på å utvikle produktet ved å komme med dine forslag til hvordan produktet ville se ut, eller synes du det var vanskelig å delta i dette arbeidet?
6. Var det problematisk for deg på noen måte å delta i utviklingen – i så fall hvordan?
7. Følte du at ledelsen din så på din rolle som bruker som viktig for utviklingen av TelMont?
8. Følte du at ledelsen i organisasjon A og i organisasjon B så på din rolle som bruker som viktig for utviklingen av TelMont?
9. Følte du at du fikk nok støtte fra ledelsen din til å delta i prosjektet?
10. Visste du hvilken støtte som fantes i organisasjonen for denne type prosjekt?
11. Tror du at produktet ble enklere å bruke siden brukerne var med på å utvikle produktet?
12. Synes du det var for mye unødvendige detaljer på produktet
  - (a) i starten av prosjektet?

(b) når produktet var ferdig?

13. Hvor fornøyd er du med det ferdige produktet?

## Tillegg E

# Svar fra e-postintervjuene med brukerne på TelMont

### E.1 Svar fra TelMont-bruker 1

#### *Spørsmål 1:*

Personlig synes jeg at mine forslag til endringer ble positivt mottatt, vi hadde flere møter underveis der vi diskuterte utviklingen av produktet og siden jeg var den eneste som hadde bakgrunn fra Organisasjon X følte jeg at mine vurderinger ang. dette fagområdet ble tatt på alvor.

#### *Spørsmål 2:*

Noen av endringene ble gjennomført, særlig husker jeg verifiseringen av numrene som bærer nødsamband, 110, 112, 113. Dette var svært viktig å få med og dette ble implementert i TelMont etter hvert. Jeg kom nok litt for sent inn i prosjektet, deltok første gang i februar 2000. Hadde jeg blitt involvert tidligere tror jeg at jeg kunne hatt større påvirkning på produktet ang. Organisasjon X-siden. Er beskjeden nok til å tro at vi kunne ha spart penger, dersom vi hadde hatt Organisasjon X-kompetanse i prosjektet på et tidligere tidspunkt.

#### *Spørsmål 3:*

Vi visste alle hele tiden at endringer kostet penger, vi fikk ofte spørsmål fra prosjektledelsen å vurdere kostnad opp mot nytte. Men når utgangspunktet var at en hele tiden skulle ha en gevinst på 18% av all kapital som ble brukt i prosjektet, følte jeg at vi måtte svelge alt for mange kameler underveis. Intensjonen med TelMont var langt høyere enn det produktet vi sitter igjen med. Tankene om at TelMont kunne bli et fullverdig kabelomleggingsprodukt som skulle takle linjeføring fra sentral til endefld. mener jeg var en riktig tanke, men økonomi stoppet oss veldig tidlig her. Ingen ting vondt sagt om person A sitt produkt, men dette ble en sideløsning i Excel og ikke en implementert del av TelMont Særlig støter vi på manglende Excel/generell PC-kunnskap når vi lanserer produktet. Føler også at den løsningen som vi sitter igjen med ang. DP-utstyrnummer bryter fullstendig med vedtatt strategi ang. utstyrnummerdisponering i hovedorganisasjonen. Dette er et av de områdene jeg ble mest skuffet over at vi ikke kunne få bedre løsning enn det vi har endt opp med.

#### *Spørsmål 4:*

Ja, udiskutabelt, jeg tror at det er ingen andre enn brukerne av systemet som har rette forutsetning til å fremme slike forslag. Jeg har skremmende eksempler fra utviklingen av andre systemer bla. XXX-ordresystemet der brukerne ikke er blitt tatt med på råd før der har vært for sent. F.eks. dersom en ab. flyttet og samtidig konverterte til ISDN måtte kunden skifte tlf.nr. selv om det var innenfor samme sentralområde. Særlig oppbygging av bilder, bildene sitt forhold til hverandre og logisk framdrift i programmet. Har fått gode tilbakemeldinger på at TelMont er enkelt å lære og lett å bruke.

*Spørsmål 5:*

Jeg likte svært godt å jobbe med TelMont, selv om det kunne bli vel mye av og til. Det er vanskelig å omsette teori til praksis, særlig å på forhånd å tenke ut alle eventualiteter og alternativer som må være med. Å være sikker på at en har tatt med alle muligheter og ikke har utelatt noe som underveis ville vise seg å være vesentlig for produktet. Det var litt som å bygge et hus, hvordan blir det som man setter på en tegning seende ut i virkelig liv. Dette ble lettere etter hvert som noe kom på plass, og jeg mener det var rett at vi delte dette opp i to leveranser. Jeg føler at det var mye lettere å jobbe med andre versjonen av TelMont enn den første. Da hadde vi et forhold til bildene og til enkel bruk av programmet, avgjørelser blei lettere å ta fordi enn kunne se konsekvenser i forhold til førsteutgaven.

*Spørsmål 6:*

Ikke direkte problematisk, men TelMont har krevd mange arbeidstimer, uten at jeg har fått noen form for avlastning i min daglige jobb. Å være på Lillehammer en uke om gangen kan jo selvfølgelig være problematisk for noen, men for min del har det vært greit. Det er mye lettere å ha fokus på oppgaven når en er fysisk borte fra sin daglige arbeidsplass. TelMont-prosjektet har hele tiden hatt for lite personalressurser, og dette har gjort sitt til at trykket på den enkelte er blitt for stort. Men hvem tenker på det nå når vi har produktet ferdig?

*Spørsmål 7:*

Min leder har hele tiden vært veldig positiv til min deltagelse i prosjektet, han har jo visst at jeg skulle være en av brukerne av produktet, og muligheten til å påvirke og til å lære produktet ville være størst ved å delta.

*Spørsmål 8:*

Har ikke hatt noen direkte kontakt med organisasjon A sin ledelse eller organisasjon B sin ledelse i forbindelse med TelMont, slik at jeg må svare nei på dette spørsmålet. Jeg mottok et takkebrev fra person B like etter jul og det er dessverre den eneste henvendelsen jeg har fått fra organisasjon A sin ledelse. Dersom en ser på organisasjon B på utviklingsplan og ikke på ledelsesside, så synes jeg at mitt forhold til organisasjon B har vært veldig nyttig, vi har utfylt hverandre på en fin måte og etter hvert har organisasjon B sin kompetanse på våre produkter som har kommet prosjektet til nytte.

*Spørsmål 9:*

En skal jo være ærlig når en svarer på slike spørsmål, og mitt utgangspunkt for å delta i prosjektet var at jobben min måtte komme i første rekke. "Bare jobben blir gjort, må du gjerne delta i prosjektet". Det blir jo til at en strekker seg ekstra langt når noe er gøy å jobbe med. Det jeg mener å si er at jeg har ikke fått nok støtte, men gjelder ikke isolert sett bare til prosjektet. Arbeidsmengde og bemanning står

ikke i riktig forhold innen mitt fagområde. Vi er to som er brukeransvarlige i Markedsområde Vest og vi dekker begge Agder-fylkene, Rogaland, Hordaland og Sogn og Fjordane.

*Spørsmål 10:*

Nei, dette er første gangen jeg deltar i sentralt prosjekt. Har tidligere deltatt i diverse lokale prosjekt. Så dette var en ny og svært nyttig erfaring for meg. Jeg føler at jeg har økt min kompetanse inne mitt fagfelt med å delta i prosjektet. Har opparbeidet gode relasjoner til mine kollegaer og personlig føler jeg at jeg kommer styrket ut av å har deltatt i prosjektarbeidet. F.eks. å knytte uformelle kontakter inne ulike fagområder i organisasjon A og hos organisasjon B, vil jeg dra stor nytte av senere, også det å kunne se mitt snevre fagområde i et større perspektiv.

*Spørsmål 11:*

Ja, dette har jeg vel svart på tidligere, men vi opplever veldig mange hvorfor ikke sånn og hvorfor ikke det, når vi er ute å demonstrerer produktet for brukerne. Her kunne vi helt sikker vært flinkere og kanskje vi skulle hatt høringer underveis for å se om vi gikk i takt med brukerne. Noen er mer kritiske enn andre og når en har jobbet lenge med et produkt så blir en mindre kritisk til det en har gjort. Jeg tror at det ville være nyttig til å få noen andre, enn vi som har vært med å utvikle produktet, til å evaluere det.

*Spørsmål 12:*

I begynnelsen følte jeg det, vi hadde mange og lange diskusjoner, der jeg følte at vi sto helt stille. Detaljene var mange og enkelte sitt markeringsbehov var stort, jeg har aldri tidligere opplevd den typen møter som vi hadde i oppstarten, og jeg lurte mange ganger på hva det var som egentlig foregikk. Etter hvert forstod at behovet for detaljer var viktig, men personlig synes jeg at min arbeidstid er for verdifull til å sitte å krangle om detaljer. Mange av diskusjonene som foregikk den gangen burde vært gjort unna på et høyere nivå. Nå når produktet er ferdig ser jeg klart behovet for alle detaljene, men ambisjonsnivået for TelMont i starten av prosjektet var noe helt annet enn det produktet vi har fått. Hadde vi helt fra starten ut av justert nivå til det som viste seg å være realistisk, hadde vi spart oss for mye tid og bekymringer.

*Spørsmål 13:*

Jeg er svært fornøyd med produktet, det er lett å lære, lett å bruke og det letter denne typen arbeidsoppgaver mye. Å kunne flytte ab. uten å måtte tenke på kundeprofil og undernr. er nesten som en drøm. Tidligere kunne en holde på med slike arbeidsoppgaver i dager og uker uten at en følte noen fremdrift og en gikk fort lei. Nå er det nesten som gleder seg til en slik jobb. Målet nå er å få spredd budskapet om TelMont, slik at vi får avkastning for millionene som er brukt. Her er det fremdeles mye å hente. Senest i forrige uke var det bestilt ab-flytting hos organisasjon Z, uten at TelMont var informert.

## E.2 Svar fra TelMont-bruker 2

*Spørsmål 1:*

Vi diskuterte endringene internt i testteamet (2 brukere) og foreslo videre dette overfor utvikler/e tidlig i fase 1.0. Det ble nært diskutert med denne/disse og vi kom ut med en optimal løsning. I senere trinn 1.5 og 2.0 er alle endringsforslag

diskutert godt gjennom på forhånd blant prosjektmedlemmene organisasjon A og også sammen med organisasjon B. Endringer og forslag på tilpasninger er møtt med åpenhet og diskutert nøye i hele prosjektet.

*Spørsmål 2:*

Ja, mange av mine foreslåtte endringer er tatt hensyn til.

*Spørsmål 3:*

I trinn 1.5 og 2.0 var dette med pengebruk mer fremtredende og organisasjon B fortalte med en gang at de ville kreve ev. penger hvis forslag var noe annet en spec tilsa. Nytteverdien ble da diskutert internt i prosjektet, før det ev. ble bestilt.

*Spørsmål 4:*

Ja, både det og for å oppfylle minstekrav for funksjonalitet.

*Spørsmål 5:*

Det var veldig tøft til å begynne med. Kommunikasjonen internt i prosjektet både mot medlemmer fra mitt selskap og mot de som skulle utvikle var krevende da de var fullstendig ukjent med et viktig støttesystem som ble benyttet.

Det var også nytt for meg å være med på dette. Oppgaven så voldsom ut og følte at jeg trengte flere fra brukersiden med i prosjektet for å ha noen å støtte meg til faglig. Etter hvert kom dette på plass og kommunikasjonen ble også enklere etter hvert. Dette gjorde at trivselsfaktoren ble stor etter hvert da ting begynte å "flyte" bedre.

*Spørsmål 6:*

Nei, men det ble til tider stort arbeidspress selv om jeg i perioder var mer eller mindre fristilt fra mine daglige gjøremål. Med min stilling, så blir man aldri fristilt 100% selv om man har lagd en ressursavtale på dette. Det måtte også god organisering til privat da det ble mye mer fravær fra hjemmet enn det ellers er.

*Spørsmål 7:*

Ja, jeg har ikke noe annet inntrykk. Jeg avsto en del møtevirksomhet og prosjektarbeid, og dette fikk jeg aksept for.

*Spørsmål 8:*

Ja, alle endringer og forslag har jeg hatt mulighet til å si min mening om. Når min løsning syntes å være den mest hensiktsmessige, så ble den valgt.

*Spørsmål 9:*

Ja, jeg er gitt frihet til å disponere tida mi i forhold til trykket i prosjektet. Hjemmekontoret er blitt benyttet mer enn ellers. Dette var det full aksept for.

*Spørsmål 10:*

Aner ikke.

*Spørsmål 11:*

Helt klart. Har bare fått positive tilbakemeldinger fra andre brukere etter at produktet er tatt i bruk.

*Spørsmål 12:*

(a)

Det er klart at det ble en god del fagprat på en rekke møter som jeg kunne vært foruten. Ellers er det viktig at man kjenner all funksjonalitet for å forstå det ferdige produktet fullt ut og for å kunne fastslå om det virker i henhold til det krav som

er satt.

(b)

Nei, men den del uvesentlige mailing underveis og etterpå har det vært.

*Spørsmål 13:*

Jeg er kjempefornøyd. Det forenkler hverdagen og gjør at man slipper og taste en million ganger hver dag. Dette reduserer nakke- og rykkskader i hele organisasjonen.

### E.3 Svar fra TelMont-bruker 3

*Spørsmål 1:*

Jeg/vi så jo ting underveis som ble ønsket som endringer, og det ble godt mottatt (i og med at det var et av elementene som vi SKULLE gjøre i prosjektet). Endringsforslag ble gjerne diskutert av oss i gruppen utenfor organisasjon B FØR det ble sendt inn.

*Spørsmål 2:*

Det var da både og. Ikke alle endringsforslag var neppe helt gjennomtenkt, eller ikke var praktisk mulig (i forb. med grensesnittproblemer etc), og ikke minst økonomi — etter hvert.

*Spørsmål 3:*

Som i forrige svar — etter hvert ble jo økonomien viktig pga. tidligere “overskridelser”, men de fleste endringene ble dog innført. Vi var hele tiden informert om det økonomiske aspektet, selv om min deltagelse i prosjektet ikke skulle la seg styre av det. Ble noe for dyrt fikk vi beskjed.

*Spørsmål 4:*

Ja — men også der i fellesskap med de andre.

*Spørsmål 5:*

Det var vanskelig i starten, da det var veldig mye papirer med diverse spesifikasjoner, samt at jeg ikke følte meg datakyndig nok til å følge med i “fagspråket” til tider.

*Spørsmål 6:*

Se forrige svar, men etter hvert ble min rolle mer definert som “ekspert” i et tilstøtende støttesystem, og da ble jeg mer på “hjemmebane”.

*Spørsmål 7:*

Ja.

*Spørsmål 8:*

Ja, etter hvert — når mitt ansvar ble definert (se svar på to spørsmål tilbake).

*Spørsmål 9:*

Ja, absolutt!

*Spørsmål 10:*

Nei, ikke ved oppstart.

*Spørsmål 11:*

Definitivt ja! Erfaringer med andre prosjekter som ikke har innbefattet brukere har det i ettertid vært mye problemer med.

*Spørsmål 12:*

Ja, det var mye detaljer som jeg drøssevis av mailer/informasjon om som det ikke var

behov for, men som hørte til “høyere opppe” i prosjektet, så som økonomidetaljer samt avtaler og så videre. Dette føler jeg gikk som en rød tråd gjennom hele prosjektet.

*Spørsmål 13:*

Så langt veldig bra fornøyd, men det er så ferskt at det ennå dukker opp en del småfeil.

## E.4 Svar fra TelMont-bruker 4

*Spørsmål 1:*

Endringer blei godt mottatt. Vi måtte sjølsagt ha ein god og vettig begrunnelse for endringsforslaget.

*Spørsmål 2:*

Dersom den økonomiske gevinsten var forsvarlig.

*Spørsmål 3:*

Ved utvikling av versjon 1 fikk man ikkje vite kva en endring kosta. Dette var meir endring av ein eksisterande spesifikasjon og ikkje endring av eit utvikla produkt/applikasjon. Endringar etter versjon 1 vart satt i produksjon vart estimert i kr.

*Spørsmål 4:*

JA

*Spørsmål 5:*

Det er nok deltaking i dette prosjektet som gjer at eg fortsatt arbeider her.

*Spørsmål 6:*

Det var vanskelig å få godkjenning av den nærmeste ledelsen til å bruke så mykje tid på eit prosjekt. Dessuten er reiseaktivitet litt problematisk når man er gift med ein sjukepleir og har småbarn.

*Spørsmål 7:*

Min rolle i prosjektet var ikkje som brukar, men som ressursperson. Eg hadde tatt fram det produktet som var i bruk før denne applikasjonen kom på lufta.

*Spørsmål 8:*

JA

*Spørsmål 9:*

Nei! Eg måtte overtale for å få lov til å delta i prosjektet.

*Spørsmål 10:*

Eg kjente til prosjektkontoret.

*Spørsmål 11:*

JA

*Spørsmål 12:*

(a)

Den opprinnelige specen var laga for eit anne sluttprodukt enn det vi ser i dag. Dermed inneholdt den meir detaljar enn nødvendig for den funksjonalitet vi trengte for trinn 1 og 2 av prosjektet.

(b)



Detaljer skaper heilhet. Eg synst vi fikk ett godt produkt, og som brukarane også synst fungerer godt.

*Spørsmål 13:*

Ja. Man blir aldri fornøgd med et dataprogram. Med bakgrunn i denne påstanden meiner eg vi har fått ett veldig bra produkt.