

INSTITUTT FOR INFORMASJONS- OG MEDIEVITENSKAP

Masteroppgave

Verdandi

- utvikling og evaluering av en digital informasjonsradiator
på en videovegg

Av: Sindre Thormodsæter Benonisen og Kjetil Bruland

Veileder: Solveig Bjørnstad

1. juni 2012

Sammendrag

Hensikten med denne studien var å utvikle og evaluere en digital informasjonsradiator for å få økt innsikt i hvordan man kan kombinere egenskaper ved fysiske informasjonsradiatorer med programvare. For å fremme synligheten til informasjonsradiatoren og for å beholde aktiviteten foran den, er et 18 skjermers videoveggssystem benyttet sammen med interaksjonsteknologien Microsoft Kinect. Gjennom tre iterasjoner ble prototypen, Verdandi, utviklet og evaluert i form av brukertester med totalt 19 deltakere. Prototypen ble avslutningsvis evaluert med et rammeverk for kognitive dimensjoner. Resultatet av studien er en egnet interaksjonsmetode til Verdandi og retningslinjer for utvikling av digitale informasjonsradiatorer.

Forord

Vi vil først og fremst takke vår veileder, Solveig Bjørnstad. Du har vist oss stor tillit, rettleidet oss når vi hadde behov for hjelp og alltid hatt en åpen dør.

Aleksander Krzywinski og Tor Gjørseter: takk for utfordringen med MIMIR, for gode råd og hjelp med teknisk utstyr. Også takk til Frode Guribye for lån av opptaksutstyr og Rune Arntsen for tilgang til fasiliteter.

Takk til Stian Benonisen Thormodsæter og, den uoffisielle kretsmester i komma, Teis Lunde Lømo for korrektur og gode tilbakemeldinger.

En stor takk til våre medstudenter på rom 638, Erlend «Alan! Alan!» Andresen og Regine Oliversen Sagstad. Takk for råd, tilbakemeldinger, gullkorn og god stemning (det er ikke lov å si (Radioresepsjonen, 2012)).

Sist, men ikke minst, vil vi takke alle deltakerne for deres hjelp og verdifulle tilbakemeldinger.

Innholdsfortegnelse

Sammendrag	ii
Forord	iii
Innholdsfortegnelse	iv
Figurer	ix
Tabeller	x
1 Introduksjon	1
2 Teori	5
2.1 Smidige artefakter	5
2.1.1 Informasjonsradiator	6
2.1.2 Brukerhistorier	7
2.1.3 Oppgaver	8
2.1.4 Burndowndiagram	9
2.1.5 En analyse av fysiske artefakt	9
2.1.6 Digitale verktøy for prosjekthåndtering	13
2.2 Videovegg	17
2.3 Interaksjon med videovegger	19
2.4 Oppsummering	21
3 Metode	22
3.1 Designforskning	22
3.2 Brukertesting	23
3.2.1 Intervju	24
3.2.2 Think-aloud	24
3.2.3 Deltakere	26
3.3 Brukbarhetsinspeksjon	26
3.3.1 Heuristisk evaluering	27
3.3.2 Kognitivt dimensjonsrammeverk	28
3.4 Oppsummering	32

4	Forskningsdesign	33
4.1	Designforskning	33
4.2	Brukertesting	34
4.3	Iterasjon 0 - Systemoppsett og valg av interaksjonsteknologier	36
4.4	Iterasjon 1 - Evaluering av interaksjonsmetoder	36
4.5	Iterasjon 2 og 3 - Utvikling og testing av Verdandi	37
4.6	Kognitiv dimensjonsanalyse	40
4.7	Oppsummering	41
5	Iterasjon 0	43
5.1	Systemoppsett og konfigurasjon av MIMIR	43
5.1.1	Windows versus Linux	44
5.2	Interaksjonsmetoder	45
5.2.1	Gyromus	46
5.2.2	Microsoft Kinect	46
5.2.3	Nintendo Wii	48
5.3	Utviklingsrammeverk	51
5.4	Oppsummering	52
6	Iterasjon 1	53
6.1	Interaksjon	53
6.2	Interaksjonstester	53
6.2.1	Knappetest	54
6.2.2	Blinktest	54
6.2.3	Test av hoverering	55
6.2.4	Biltest	55
6.2.5	Gesttest	55
6.3	Brukertest	56
6.4	Resultat	56
6.4.1	Knappetest	57
6.4.2	Blinktest	58
6.4.3	Blinktest med mindre blink	59
6.4.4	Test av hoverering	60
6.4.5	Biltest	60
6.4.6	Gesttest	61

6.5	Diskusjon	62
6.5.1	Refleksjoner rundt testene	64
6.5.2	Aktiviteten og synlighet	65
6.6	Oppsummering	65
7	Iterasjon 2	66
7.1	Krav	66
7.2	Prototype	70
7.2.1	Struktur	70
7.2.2	Brukerhistorier og oppgavekort	71
7.2.3	Generell informasjon	72
7.2.4	Burndowndiagram	72
7.2.5	Kildekontroll- og informasjonsmeldinger	73
7.2.6	Interaksjon	73
7.3	Brukertest	74
7.3.1	Deltakere	74
7.3.2	Introduksjon og intervju	74
7.3.3	Opplæringsfase	75
7.3.4	Test	75
7.4	Koding	75
7.5	Analyse	76
7.5.1	Deltakernes kunnskap	76
7.5.2	Informasjon, burndowndiagram og meldinger	79
7.5.3	Brukerhistorie- og oppgavekort	81
7.5.4	Interaksjon	83
7.5.5	Utvidelser	84
7.5.6	Kravkatalog	85
7.6	Diskusjon	85
7.6.1	Deltakernes kunnskap	85
7.6.2	Informasjon, burndowndiagram og meldinger	85
7.6.3	Brukerhistorie- og oppgavekort	86
7.6.4	Interaksjon	87
7.6.5	Refleksjoner rundt iterasjonen	87
7.7	Oppsummering	88

8	Iterasjon 3	89
8.1	Prototype	89
8.1.1	Struktur	89
8.1.2	Brukerhistorie- og oppgavekort	90
8.1.3	Burndowndiagram	91
8.1.4	Sprintinformasjon	92
8.1.5	Brukerinformasjon	93
8.1.6	Commit- og informasjonsmeldinger	93
8.1.7	Ansattliste	93
8.1.8	Animasjon av ferdige brukerhistorier	94
8.1.9	Interaksjon	94
8.1.10	Navigasjonsmeny	94
8.1.11	Navigasjonsindikator	95
8.2	Brukertest	95
8.2.1	Deltakere	96
8.2.2	Introduksjon og intervju	96
8.2.3	Opplæring	96
8.2.4	Test	96
8.3	Koding	97
8.4	Analyse	98
8.4.1	Deltakernes kunnskap	98
8.4.2	Informasjon, burndowndiagram og meldinger	99
8.4.3	Brukerhistorier og oppgaver	100
8.4.4	Interaksjon	100
8.4.5	Utvidelser	101
8.5	Diskusjon	102
8.5.1	Deltakernes kunnskap	102
8.5.2	Informasjon, burndowndiagram og meldinger	102
8.5.3	Brukerhistorier og oppgaver	103
8.5.4	Interaksjon	104
8.5.5	Utvidelser	105
8.6	Oppsummering	106
9	Kognitiv dimensjonsanalyse	108
9.1	Dimensjonene i forhold til Verdandi	108

9.2	Diskusjon	113
9.2.1	Dimensjonenes relasjoner og designmanøvre	114
9.2.2	Mediets og interaksjonenmetodens påvirkning	114
9.2.3	Egenskaper ved en digitalisering	116
9.2.4	Konfigurasjon og tilpassing	117
9.2.5	Applikasjonen som utvikles	118
9.3	Oppsummering	120
10	Konklusjon	123
10.1	Refleksjoner rundt studien	125
10.2	Videre forskning	126
10.3	Avslutningsvis	127
	Referanser	128
	Vedlegg A Kravkatalog for iterasjon 2	134
	Vedlegg B Kravkatalog til iterasjon 3	135
	Vedlegg C Kategoriskjema til koding av brukbarhetsproblemer	136
	Vedlegg D Mal for brukertest - Iterasjon 1	137
	Vedlegg E Mal for brukertest - Iterasjon 2	142
	Vedlegg F Mal for brukertest - Iterasjon 3	145
	Vedlegg G Sammenligning av kognitive dimensjoner	149
	Vedlegg H Prototype iterasjon 2	152
	Vedlegg I Prototype iterasjon 3	153

Figurer

1	Informasjonsradiator	2
2	MIMIR	2
3	Kategorisering av verktøy.	13
	(a) Kategorier av verktøy i samlokaliserte eller distribuerte team.	13
	(b) Overlapp mellom fysiske og smidige verktøy.	13
4	Aspekter ved ulike verktøy.	14
	(a) De mest tilfredsstillende aspektene.	14
	(b) De minst tilfredsstillende aspektene.	14
5	Egenskaper som behøves i verktøy.	14
6	Oppsett av laboratoriemiljø.	35
	(a) Laboratoriemiljøet sett ovenfra.	35
	(b) Utnsnitt fra videokamera.	35
7	Transkript før og etter metainformasjon.	39
8	XQuery-uttrykk for å slå sammen og sortere brukbarhetsproblem.	40
9	Forskningsdesign.	42
10	FAAST kalibrering	47
11	Interaksjonsmetoder med Kinect.	48
	(a) Illustrerer hvordan man velger med pushmetoden.	48
	(b) Illustrerer hvordan man velger med pingvinklikkmetoden.	48
12	Nintendo Wii: Dekningsgrad.	50
13	Knappetestens utforming.	54
	(a) Knappetestens utgangspunkt.	54
	(b) Knappetesten underveis.	54
14	Blinktestens utforming.	55
	(a) Blinktestens utgangspunkt.	55
	(b) Blinktesten med resultat.	55
15	Biltestens utforming.	55
	(a) Biltesten ved korrekt utførelse.	55
	(b) Biltesten ved utforkjøring.	55
16	Gesttestens utforming.	56
	(a) Gesttestens utgangspunkt.	56
	(b) Gesttestens utforming etter to gester.	56

17	Resultat fra knappetest.	57
	(a) Interaksjonsteknologienes nøyaktighet.	57
	(b) Interaksjonsteknologienes tidsbruk.	57
18	Ulike interaksjonsmetoders presisjon, tidsbruk og kontrollfølelse.	58
19	Resultat av blinktest.	59
20	Resultat av biltest.	61
21	Resultat av gesttest.	62
22	Verdandi i iterasjon 2.	70
23	Verdandi med skjermrammer i iterasjon 2.	71
24	Brukerhistoriekort i iterasjon 2.	71
25	Oppgavekort i iterasjon 2.	72
26	Burndowndiagram i iterasjon 2	73
27	Deltakers informasjonsradiator (Pirates)	77
28	Deltakers informasjonsradiator(Goblins)	78
29	Deltakers informasjonsradiator (Blue)	79
30	Interaksjonsmetodens presisjon i iterasjon 2.	83
31	Verdandi i iterasjon 3	89
32	Markering av brukerhistorie i iterasjon 3.	90
33	Brukerhistoriekort i iterasjon 3.	91
34	Oppgavekort i iterasjon 3.	91
35	Burndowndiagram iterasjon 3.	92
36	Ansattliste, commit- og informasjonsmeldinger.	93
37	Navigasjonsmeny	95
38	Navigasjonsindikator	95
39	Relasjon mellom funn og retningslinjer	122

Tabeller

1	Eksempler på digitale nett- eller programvarebaserte informasjonsradiatorer.	7
2	Identifiserte fordeler og ulemper ved ulike verktøy.	16

1 Introduksjon

Every attempt I've seen to computerise stories has failed to provide a fraction of the value of having real cards on a real wall.

— Beck & Andres (2004, s. 45)

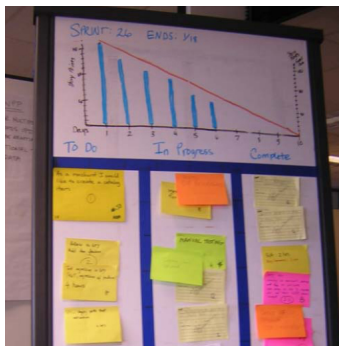
Introduksjonen av smidige systemutviklingsmetoder førte til en endring i hvordan programvare blir utviklet. Fokuset på mennesker, samarbeid, kommunikasjon og mindre dokumentasjon krevde nye måter å planlegge og koordinere på. Ut fra dette fokuset kom ideen om den informative arbeidsplassen, en praksis med mål om å enkelt visualisere fremgangen til et prosjekt i lokalet hvor utviklerne arbeider. Sentralt i praksisen er informasjonsradiatoren, et artefakt med mål om å synliggjøre og fasilitere den smidige utviklingsprosessen (Cockburn, 2002).

Begrepet informasjonsradiator brukes om artefakter som burndowndiagram, oppgavetavle, scrumvegg/scrumtavle og andre artefakter som synliggjør og fasilitere den smidige utviklingsprosessen (Perry, 2008). En av de viktigste informasjonsradiatorene er oppgavetavlen (Cockburn, 2002), som kan kombineres med andre informasjonsradiatorer som burndowndiagram (Perry, 2008). I denne studien brukes begrepet informasjonsradiator i hovedsak for å beskrive en oppgavetavle, men blir også brukt for å dekke systemer som bruker en kombinasjon av artefakter. Et eksempel på en informasjonsradiator illustreres i figur 1. Informasjonsradiatorer visualiserer progresjon ved at utviklere interagerer med og flytter rundt på kartotek kort eller post-it-lapper, som representerer brukerhistorier. Artefaktene, informasjonsradiator og brukerhistoriekort, har gjentatte ganger blitt digitalisert som programvare, men likevel er bruken av fysiske artefakter svært utbredt (Azizyan, Magarian & Kajko-Matsson, 2011).

Motivasjonen og inspirasjonen til denne masteroppgaven stammer fra en studie utført av Sharp, Robinson & Petre (2009). For å bedre forstå rollen til fysiske informasjonsradiatorer og kort, analyserte Sharp et al. (2009) artefaktene fra et notasjonelt og sosialt perspektiv. Det notasjonelle perspektivet i studien vurderer implikasjonene artefaktene har for kognisjon, samt hvordan artefaktene brukes og forstås. Det notasjonelle perspektivet eksisterer ikke i isolasjon, det brukes i en sosial kontekst der sosiale konvensjoner og praksiser vil påvirke bruken av artefaktene. Sharp et al. (2009) konkluderer med at det notasjonelle og sosiale

perspektivet er gjensidig støttende, og at rollen og aktiviteten ved artefaktene må forstås for å skape digitale ekvivalenter.

I denne masteroppgaven vil det bli utviklet en digital informasjonsradiator som baserer seg på den smidige metoden Scrum og som bevarer aktiviteten foran informasjonsradiatoren. Terminologi og artefakter fra Scrum benyttes på grunn av tilgjengelighet av informanter, som har kjennskap til metoden, både internt på Universitetet i Bergen og i lokale IT-bedrifter. For å fasilitere aktiviteten vil prototypen utvikles på MIMIR¹, en videovegg som består av 18 skjermer, og med interaksjonsmetoder som ivaretar synligheten ved aktiviteten. Ved å benytte MIMIR fremfor projektor vil informasjonsradiatoren ha et svært høyt detaljnivå, hvilket vil gjøre det mulig å vise mer informasjon. MIMIR er illustrert i figur 2.



Figur 1: viser et eksempel på en informasjonsradiator som bruker kolonneinndeling, postit-lapper og burndowndiagram. Figuren er hentet fra Perry (2008).



Figur 2: viser hvordan MIMIR er satt opp med 18 skjermer som alle er 24 tommer, med en aluminiumsramme som holder skjermene sammen.

Ved å digitalisere informasjonsradiatoren oppnås en rekke fordeler, siden digitale system kan automatisere en rekke prosesser og handlinger. Blant annet kan en digitalisering forenkle oppgaver som rapportering, deling av informasjon mellom distribuerte team og integreres med andre systemer.

En digitalisering av informasjonsradiatoren vil trolig forandre notasjonsperspektivet, som Sharp et al. (2009) har analysert med det kognitive dimensjonsrammeverket som ble utviklet av Green i 1989. Ved å bruke det samme rammeverk vil det være mulig å sammenligne om, og hvordan, dimensjonene endres. Første forskningsspørsmål er:

¹MIMIR: Multimodal Interactive Multimedia Information Rig

FS1: Hvordan endres notasjonsperspektivet ved informasjonsradiatoren når den digitaliseres på MIMIR?

Ved å besvare FS1 kan det gis økt innsikt i hva som endres når informasjonsradiatoren digitaliseres, og kan være med på å forklare hvorfor blant annet Beck & Andres (2004) hevder at fysiske artefakt er overlegne i forhold til programvare. For å besvare FS1 er en prototype, kalt Verdandi, iterativt utviklet og brukertestet slik at en digital informasjonsradiator kan evalueres med det kognitive dimensjonsrammeverket.

Aktiviteten som skjer foran den fysiske informasjonsradiatoren er synlig for alle i rommet, fordi brukerne må reise seg opp og fysisk interagere med informasjonsradiatoren. Denne aktiviteten skjules hvis brukere sitter på egne arbeidsstasjoner og oppdaterer et regneark eller annen programvare. Flere interaksjonsmetoder kan brukes for å etterligne den fysiske aktiviteten. Forsknings spørsmål to lyder derfor som følger:

FS2: Hvilken interaksjonsmetode er best egnet i kombinasjon med MIMIR for å bevare aktiviteten ved en informasjonsradiator?

Intensjonen med FS2 er å finne en interaksjonsmetode som oppfyller krav om presisjon, effektivitet og enkelhet, og samtidig bevarer den fysiske aktiviteten foran informasjonsradiatoren. For å besvare FS2 brukertestes potensielle interaksjonsmetoder for å finne den best egnede kandidaten. Interaksjonsmetoden som er best egnet vil brukes sammen med prototypen i påfølgende iterasjoner for å verifisere at metoden som velges oppfyller kravene og fungerer til formålet.

FS1 og FS2 kan belyses fra ulike perspektiv. Siden forsknings spørsmålene i stor grad handler om interaksjon med programvare, kan metoder innenfor Human Computer Interaction (HCI) som for eksempel grounded theory, distribuert kognisjon og aktivitetsteori være naturlig å anvende i denne studien. Felles for disse metodene er at de vanligvis brukes for å strukturere analysen av data som er samlet fra feltstudier (Sharp, Rogers & Preece, 2011). Siden det ikke eksisterer et ferdig artefakt som kan evalueres i et naturlig miljø, vurderes disse metodene som lite egnet for å besvare forsknings spørsmålene. Av samme årsak utelukkes metoder innenfor datastøttet samarbeid², da det ikke eksisterer

²Eng.: Computer Supported Cooperative Work

et artefakt som er modent nok for å evalueres som en gruppeaktivitet. For å kunne svare på forskningsspørsmålene må det utvikles en prototype, og derfor velges designforskning som overordnet metode. Utviklingen av prototypen utvikles iterativt som en del av designsyklusen i designforskning. Prototypen utvikles på bakgrunn av krav som utformes i forhold til teori og tidligere forskning på *artefakter* innenfor smidige metoder, samt videovegger og relaterte interaksjonsmetoder.

En avgrensning i utformingen av prototypen er at det fokuseres på den daglige aktiviteten rundt informasjonsradiatoren, og inkluderer ikke planleggingsfasen der brukerhistoriene opprettes. Digitale artefakt som fokuserer på planleggingsfasen er tidligere utviklet, eksempelvis har Wang & Maurer (2008) utviklet «AgilePlanner», et horisontalt berøringsbasert skjermssystem som tar for seg opprettelse og planlegging av brukerhistorier.

Ved hver iterasjon i designsyklusen evalueres prototypen med brukertester. Deltakerne i brukertestene velges på bakgrunn av tidligere kjennskap til smidige metoder og informasjonsradiatorer. Hver brukertest består av et intervju og en guidet observasjon med think-aloudteknikken. Datamaterialet fra testene blir transkribert, kodet og kategorisert for å avdekke brukbarhetsproblemer ved prototypen. Brukbarhetsproblemene brukes som utgangspunkt for videre utvikling, og prototypen evalueres avslutningsvis med en brukbarhetsinspeksjon.

Resten av oppgaven er strukturert ved at det innledningsvis i kapittel 2 presenteres teoretisk bakgrunn innenfor de relevante områder for studien. Kapittel 3 beskriver metodologi for forskning og forskningsdesignet i kapittel 4 beskriver den metodiske utførelsen av studien. Kapittel 5 gir en detaljert beskrivelse av systemoppsettet til MIMIR, samt vurderinger av interaksjonsteknologier og avslutningsvis teknologier og rammeverk. Kapittel 6 tar for seg brukertesting og evaluering av aktuelle interaksjonsmetoder. Kapittel 7 og 8 beskriver iterasjonene med utvikling, brukertesting og evaluering av prototypen, samt en analyse og diskusjon av hver iterasjon. Kapittel 9 tar for seg en brukbarhetsinspeksjon av prototypen med det kognitive dimensjonsrammeverket, resultatet av inspeksjonen vurderes mot funnene til Sharp et al. (2009), faglitteratur og brukertestene av Verdandi. Konklusjon, refleksjon over studien og forslag til videre forskning vil avslutningsvis presenteres i kapittel 10.

2 Teori

I dette kapittelet vil det presenteres relevant teori og tidligere forskning innenfor smidige artefakter, videovegger og interaksjon. Seksjonen smidige artefakter tar for seg hvordan artefaktene informasjonsradiatorer, brukerhistorier, oppgaver og burndowndiagram fremstilles i faglitteratur og forskningsartikler. Videre presenteres artikkelen til Sharp et al. (2009), som var inspirasjon til denne studien. Seksjonen om smidige artefakter avsluttes ved å presentere tre studier som tar for seg egenskapene til fysiske og digitale artefakter innenfor smidige metoder. Den neste seksjonen, videovegger, tar for seg ulike definisjoner, typer, bruksområder, fordeler og utfordringer med videovegger. Avslutningsvis følger en seksjon som tar for seg studier som har utviklet og testet ulike interaksjonsmetoder for interaksjon med videovegger.

2.1 Smidige artefakter

Smidige utviklingsmetoder har i løpet av det siste tiåret hatt en innvirkning på hvordan programvare utvikles. Nøkkelkonseptene definert i det agile manifestet³ konkretiserer egenskapene ved flere smidige metoder som eXtreme Programming (XP), Crystal familien, Feature Driven Development, Rational Unified Process, Dynamic Systems Development og Scrum (Abrahamsson, Salo, Ronkainen & Warsta, 2002). Smidige metoder ses på som en reaksjon på tradisjonelle metoder. Tradisjonelle metoder er rasjonelle og planbaserte, mens smidige metoder adresser utfordringene i en uforutsigbar verden og vektlegger verdien som kompetente mennesker og deres relasjoner gir til programutvikling (Dyba & Dingsoyr, 2009).

I følge Brown, Lindgaard & Biddle (2008) har artefakter lenge blitt brukt i design og utvikling av programvare, og artefakter er brukt som en basis for å organisere diskusjoner og samarbeid. De smidige metodene introduserer nye artefakter og Brown et al. (2008) eksemplifiserer flere av disse artefaktene, deriblant brukerhistorier og informasjonsradiatorer. Listen utvides av Schwaber (2004) med Scrum-artefaktene burndowndiagram og produkt- og sprintbacklog.

Med unntak av informasjonsradiatorer foreligger det i følge Brown et al. (2008)

³Det agile manifest: <http://agilemanifesto.org>

lite teori om hvordan smidige artefakter fungerer i praksis. Følgende seksjon vil belyse ytterligere teori om informasjonsradiatorer og dens hovedelementer som er brukerhistoriekort, oppgavekort, burndowndiagram. Avslutningsvis presenteres en studie av informasjonsradiatoren og kortenes rolle, samt tre studier som ser på egenskaper ved fysiske og digitale artefakt innenfor smidige metoder.

2.1.1 Informasjonsradiator

En informasjonsradiator er et artefakt som benyttes av smidige utviklingsteam som en del av det Beck & Andres (2004) beskriver som en informativ arbeidsplass. Begrepet informativ arbeidsplass beskrives som en praksis, og handler om å lage en arbeidsplass som støtter opp om det arbeidet som skal utføres. En informasjonsradiator er et artefakt som benyttes for å støtte utviklingsarbeid ved å vise brukerhistorier, oppgavekort, burndowndiagrammer og andre nyttige elementer (Sharp et al., 2009). Ved å se på informasjonsradiatoren skal en interessert observatør kunne få et generelt overblikk over fremdriften i et prosjekt i løpet av 15 sekunder (Beck & Andres, 2004).

Sharp, Robinson, Segal & Furniss (2006) beskriver informasjonsradiatoren som et sted hvor brukerhistorier er strukturert etter hvor i prosessen de er kommet. Ulike utviklingsteam benytter ofte ulike strukturer hvor prosessen er delt inn i flere kolonner med brukerhistorier som skal gjøres, er under utvikling, skal testes og er ferdige (Sharp et al., 2006; Perry, 2008). I tillegg til å være et sted hvor interessenter kan oppsøke informasjon, er informasjonsradiatoren et verktøy som kan benyttes i ulike deler av utviklingen. Sharp et al. (2009) og Perry (2008) skriver at møter kan holdes foran informasjonsradiatoren og i den sammenheng fungere som et støtteverktøy.

Sharp et al. (2009) betegner informasjonsradiatoren som «veggen», og skriver at informasjonsradiatoren ofte tar form på et whiteboard, kabinettskap eller annet ledig vertikal flate. Sharp et al. (2009) argumenterer for at informasjonsradiatoren skal være et fysisk medie, og refererer til Whittaker og Schwarz som hevder at når informasjonsradiatoren er et fysisk medie blir hele teamet konstant oppmerksom på andres arbeid og at man samtidig får en grundigere refleksjon ved å manuelt manipulere et artefakt. Fysiske informasjonsradiatorer krever minimalt med trening, og dens funksjon er opplagt (Perry, 2008). Sharp et al. (2009) skriver videre at fysiske representasjoner er mer tilgjengelig enn digitale informasjonsradiatorer, og at det er vanskeligere å ignorere

den fysiske informasjonsradiatoren, spesielt hvis den endres ofte. I følge Perry (2008) har fysiske informasjonsradiatorer to begrensninger: de er kun synlige for samlokaliserte team, og de tar ikke vare på historikken til prosessen.

Beck & Andres (2004) hevder at alle forsøk, som de har kjennskap til, på å lage digitale representasjoner av brukerhistorier og informasjonsradiatorer har feilet, og at disse ikke gir en brøkdel av nytteverdien til en fysisk representasjon. Denne påstanden støttes av Cockburn (2002) som skriver at informasjonsradiatorer bør plasseres i korridorer, ikke på nettsider. Cockburn (2002) skriver videre at det krever mer innsats å åpne en nettside enn det de fleste er villige til, og at informasjonsradiatoren dermed blir skjult. Perry (2008) hevder at digitale informasjonsradiatorer fungerer bra ved distribuerte team, når informasjonen må deles, for å integrere eksisterende verktøy som bugsporingssystemer og ta vare på historisk informasjon for å identifisere trender og mønstre i teams arbeid. Noen eksempler på nettbaserte informasjonsradiatorer presenteres i tabell 1.

Atlassian Jira	Nettbasert oppgavehåndtering- og prosjektstyringsverktøy. Integrerer koblinger mot kildehåndteringsverktøy, wiki og lignende systemer som brukes for å synliggjøre applikasjonen. Støtter smidige prinsipper gjennom tilleggsmoduler som GreenHopper.
Trello	Nettbasert lettvekts oppgavehåndteringsverktøy. Oppgaver plasseres i forhold til hvor de er i prosessen.
Scrumworks	Programvarebasert smidig prosjektstyringsverktøy.

Tabell 1: viser noen eksempler på digitale nett- eller programvarebaserte informasjonsradiatorer.

2.1.2 Brukerhistorier

Når et informasjonssystem skal utvikles, har det i tradisjonelle utviklingsmetoder blitt benyttet kravspesifikasjoner for å beskrive hva systemet skal gjøre. Beck & Andres (2004) hevder det blir feil å bruke ordet «krav», siden det defineres i ordbøker som noe obligatorisk, og forbindes med noe som er absolutt og endelig, uten mulighet for endring. Som en av praksisene i XP anbefaler Beck & Andres (2004) heller å beskrive funksjonalitet som brukerhistorier. Cohn (2004) hevder at en brukerhistorie skal beskrive funksjonalitet som vil være av verdi, enten for en bruker eller en kjøper av et system eller programvare. Brukerhistorier skal gis korte navn, en skriftlig eller grafisk beskrivelse og skrives på kartotek kort sammen med et estimat av arbeidsmengden (Beck & Andres, 2004). Cohn (2004) foreslår følgende mal for å skrive brukerhistorier: som en <type bruker>, vil jeg <et

mål> slik at <en grunn>. Brukerhistorier har ifølge Jeffries (2001) tre aspekter som refereres til som de tre C-ene⁴: kortet, samtalen og bekreftelsen. I følge Jeffries (2001) skal kortet ikke inneholde all informasjon som kreves for å beskrive kravet, men skal ha akkurat nok informasjon til å identifisere, samt å gi alle involverte en påminnelse om, hva kravet omhandler.

Jeffries (2001) sin beskrivelse av kortenes innhold støttes av Rachel (gjengitt av Cohn (2004)) som hevder at et kort skal representere, men ikke detaljert beskrive, en brukerhistorie. Kortet kan inneholde en tekstlig beskrivelse brukerhistorien, mens detaljene blir utarbeidet under samtalen og lagret i bekreftelsen (Cohn, 2004). Jeffries (2001) beskriver samtalen som en utveksling av tanker, meninger og følelser. Samtalen beskrives som hovedsaklig muntlig, men kan bli supplert med dokumenter. Videre skriver Jeffries (2001) at uavhengig av hvor mye diskusjon eller dokumentasjon man produserer, kan man ikke være sikker på at brukerhistorien er ferdig implementert. For å forsikre at brukerhistorien er ferdig implementert trengs det bekreftelse, som ifølge Jeffries (2001) realiseres i form av akseptansetester.

I følge Sharp et al. (2009) og Beck & Andres (2004) er brukerhistoriekort som regel representert i form av kartotek kort, og er ofte ikke større enn 13x18 centimeter. Alle teamene i studien til Sharp et al. (2009) skrev brukerhistorier med naturlig språk, og brukte Cohns mal for å formulere brukerhistorien. Sharp et al. (2009) beskriver livssyklusen til et brukerhistoriekort med følgende steg: brukerhistorien blir skrevet på et kort, prioritert av kunden, estimert av en utvikler, plassert i en iterasjon, implementert av en utvikler, og til slutt satt som ferdig etter å ha passert en akseptansetest. For hvert steg i livssyklusen ble kortet gitt annoteringer. Annoteringene varierte mellom de ulike teamene, men forsøkte å vise den samme informasjonen: hvor lenge det vil ta å implementere brukerhistorien, hvor lang tid som har blitt brukt, hvem som har jobbet på den og brukerhistoriens nåværende status (Sharp et al., 2009).

2.1.3 Oppgaver

Beck & Andres (2004) anbefaler at utviklingsteamet deler brukerhistorier opp i oppgaver, hver med et omfang som enkeltindivider kan ta ansvar for. I følge Cohn (2004) er det to argumenter for å dele opp brukerhistorier. For det første vil det ofte være flere utviklere som

⁴De tre C-ene: Card, Conversation og Confirmation

jobber med en brukerhistorie, og brukerhistoriene kan deles opp etter teknologier som utviklerne er spesialisert innenfor. For det andre beskriver brukerhistorier funksjonalitet fra kundens perspektiv, og er ikke en liste med konkrete arbeidsoppgaver for utviklerne.

Oppgaver opprettes i XP under en planleggingsfase som har følgende steg: kunden presenterer en brukerhistorie, programmerere identifiserer oppgaver som kreves for å implementere brukerhistorien, og til slutt signerer og estimerer medlemmer av teamet oppgavene (Jeffries, 2001; Wake, 2002; Beck & Andres, 2004). Oppgaver bør skrives på egne oppgavekort (Wake, 2002) og skilles fra brukerhistoriekort med ulike farger (Sharp et al., 2009).

2.1.4 Burndowndiagram

Burndowndiagram er et artefakt som viser mengden av gjenstående arbeid i forhold til tid, og er en måte å visualisere fremgangen i et prosjekt (Schwaber, 2004). Burndowndiagram er i følge Cohn (2004) mest nyttig når de er store og synlige. Schwaber (2004) forklarer at en trendgraf i diagrammet kan benyttes for å estimere når iterasjonen vil være ferdig. Schwaber (2004) poengterer at man ved å legge til eller å fjerne funksjonalitet vil kunne se hvordan dette påvirker ferdigstillelsesdatoen. Burndowndiagrammet kan vises på en informasjonsradiator for å gi ulike perspektiver av progresjonen (Perry, 2008).

2.1.5 En analyse av fysiske artefakt

Sharp et al. (2009) utførte en studie for å bedre forstå rollen til fysiske informasjonsradiatorer og kort. De vurderte artefaktene fra et notasjonelt og sosialt perspektiv. Det notasjonelle perspektivet i studien omhandler implikasjonene artefaktene har for kognisjon, samt hvordan artefaktene brukes og forstås. Det sosiale perspektivet omhandler samspillet i et team med en sosial kontekst og spilleregler.

Sharp et al. (2009) vurderte informasjonsradiatoren og kortene i form av 14 kognitive dimensjoner. De 14 dimensjonene er en del av det kognitive dimensjonsrammeverket som ble nevnt i introduksjonen. Rammeverket og dimensjonene blir nærmere beskrevet som en del av metodekapittelet i seksjon 3.3.2. Sharp et al. vurderte begge artefaktene fra både et prosessperspektiv og et applikasjonsperspektiv. Prosessperspektivet omhandler dimensjonene i forhold til utviklingsprosessen, og applikasjonsperspektivet omhandler dimensjonene i forhold til selve applikasjonen som utvikles. Vurderingen som Sharp et al.

(2009) har utført på den fysiske informasjonsradiatoren og de fysiske kortene i forhold til de kognitive dimensjonene presenteres her:

Viskositet: Annoteringen som blir brukt på brukerhistoriekort er svært fleksibel og uttrykker en lav viskositet. Fra både applikasjons- og prosesskontekst er kortene tilgjengelige, lette å endre, fleksible og disponible. Mangelen av eksplisitte avhengigheter gjør det uproblematisk å fjerne, endre og bytte ut et kort. Annoteringer og små endringer kan enkelt bli lagt til et individuelt kort så lenge ikke meningen ved kortet endres. Å gjøre større endringer ved notasjonen, som for eksempel å endre fargekodingen, medfører repetisjonsarbeid⁵ og øker derfor viskositeten ved kortene. Informasjonsradiatorens viskositet er lav, da det er enkelt å flytte og fjerne brukerhistoriekort, selv om det kan føre til konsekvensviskositet ved at man må å flytte andre kort for å gi plass.

Synlighet: Synligheten til brukerhistoriene og informasjonsradiatoren er generelt høy, da intensjonen er at de skal være fullt synlige. Både informasjonsradiatoren og noen av annoteringene på kortene er designet for å gjøre progresjonen i iterasjonen synlig. Til og med at et kort er borte fra informasjonsradiatoren har en signifikant betydning ved at historien er under utvikling. Hverken informasjonsradiatoren eller kortene gjør applikasjonen synlig, da den manifesteres i selve koden.

For tidlige forpliktelser: Utvikling av kode basert på en brukerhistorie eller oppgave avhenger av en diskusjon og en utforskning. For tidlige forpliktelser vurderes derfor ikke som et problem når man snakker om selve applikasjonen, siden ingen forpliktelser er tatt før koden er skrevet. Det er også mulig å endre koden i ettertid gjennom refaktoreringer eller videre utvikling.

Skjulte avhengigheter: Avhengighetene til kortene er som regel skjulte når man ser dem isolert. Når kortene er på informasjonsradiatoren, blir noen avhengigheter til applikasjonen og prosessen fanget opp av de notesjonelle konvensjonene på informasjonsradiatoren. Avhengighetene blir fanget opp selv om de ikke er eksplisitt uttrykt. Informasjonsradiatoren viser hvordan kortene er i forhold til prosessen ved at kortene representerer hvilket arbeid som skal utføres i løpet av iterasjonen. Typisk vil det forekomme at funksjonelle avhengigheter blir uttrykt mellom oppgavekort og brukerhistorier, eksempelvis ved at oppgaver er plassert rett under tilhørende brukerhistorie.

⁵Eng.: repetition viscosity

Uttrykksfulle roller: Uttrykkskraften til en brukerhistories rolle er lav, siden brukerhistorien inneholder lite informasjon om dens rolle i forhold til den videre utviklingen. Det eneste som henter til selve betydningen er <en grunn> delen av en brukerhistorie fra Cohns mal, beskrevet i seksjon 2.1.2. Rollene til brukerhistorie- og oppgavekort kan skilles fra hverandre ved å eksempelvis bruke ulike farger. Informasjonsradiatoren kan øke denne dimensjonen i forhold til prosessperspektivet ved at kort innenfor en iterasjon er gruppert sammen, og videre gruppert i forhold til utviklingsprosessen. Oppgavekort er ofte lokalisert i forhold til tilhørende brukerhistorier.

Utsatt for feil: Selve notasjonen til kortene har ingen sikkerhetsmekanismer, og er derfor utsatt for feil. Kortene kan ha hvilke som helst markeringer, rekkefølge og farge. På samme måte har informasjonsradiatoren ingen sikkerhetsmekanismer, da prinsippene om organisering enten kan fastholdes eller kompromitteres.

Abstraksjon: Brukerhistorier er abstraksjoner av brukerkrav med ulike nivå, der nivået er relatert til hvor lang tid brukerhistorien vil ta å implementere. Videre beskrives abstraksjonsnivået til brukerhistorier som fleksibelt i applikasjonstermer: episke brukerhistorier⁶ har høy abstraksjon, mens oppgaver har relativt lav abstraksjon. I prosesser har brukerhistorie- og oppgavekort lavt abstraksjonsnivå. Informasjonsradiatoren gir kun enkle abstraksjoner ved hjelp av kategorisering.

Sekundærnotasjon: Hverken kortene eller informasjonsradiatoren er komplette notasjonelle system, siden de er avhengige av naturlig språk for å gjengi meningen til kravet, og av den sosiale prosessen for å tolke semantikken til informasjonsradiatoren og kortene. De kan også utvides, hvert team kan legge til notasjonelle elementer som farge, klistremerker og spesielle koder som kan utvide notasjonen.

Relasjonsnærhet: Relasjonsnærheten ved brukerhistoriekort er høy, da brukerhistorier er skrevet av kunden med naturlig språk, noe som medfører at de kobles tett mot problemdomenet. For å forstå hva en brukerhistorie betyr for applikasjonen kreves en diskusjon mellom utvikler og kunde, men utgangspunktet for diskusjonen er nært knyttet til applikasjonen. Det kreves derfor lite for å forstå kortet, men forståelsen av kortet avhenger av den sosiale konteksten. Prosessnotasjonen er nært relatert til den faktiske prosessen, da notasjonen fanger informasjon som status, hvilken utvikler som jobber på en brukerhistorie og akseptanse. Det er ingen standard på hvordan prosessnotasjonen ser

⁶Eng.: epics

ut, derfor må notasjonen først læres. Informasjonsradiatoren er hovedsaklig en innretning for å følge progresjonen, og uttrykker ikke brukerhistoriene eller applikasjonen som er under utvikling. Informasjonsradiatoren bruker struktur, layout og merkelapper for å gi en oversikt over progresjonen i en iterasjon. Notasjonen er enklere enn ved kortene, og baserer seg på ukompliserte organiseringsprinsipp. Vanligvis vil plasseringen til et kort si noe om progresjonen til brukerhistorien eller oppgaven, informasjonsradiatoren er derfor nært relatert til den faktiske prosessen, men ikke til applikasjonen.

Konsistent: Cohns mal for brukerhistorier er konsistent og enkel å bruke. Det samme gjelder annoteringene som brukes for å følge progresjonen. Når man først forstår annoteringene som brukes av teamet, er det enkelt å tyde innholdet til et kort. Meningen og detaljene kan være vanskelig å forstå for noen som ikke er kjent med XP-utvikling, men relasjonsnærhet kan muligens føre til at de som er kjent med utviklingsmetoden kan forstå et ufamiliært sett med annoteringer. Strukturen, layouten og begrepene som blir brukt på informasjonsradiatoren er konsistente innad i teamet, men kan variere fra team til team.

Diffus: Notasjonen som blir brukt på kortene er konsis/ikke diffus, og hvert element uttrykker en signifikant mening. Enkle symboler og annoteringer for å oppdatere progresjonen underbygger at denne notasjonen er konsis. Naturlig språk som brukes for å beskrive brukerhistoriene kan uttrykke en kompleks mening med få ord, og kan i denne sammenheng vurderes som konsis. Naturlig språk er derimot flertydig, og det kan være uklart hvordan innholdet tolkes av ulike brukere. Hvor konsis eller uklart en brukerhistorie er, avhenger av hvor bra eller hvor disiplinert et kort er skrevet.

Vanskelige mentale operasjoner: Brukerhistoriekortene og informasjonsradiatorens enkelhet i forhold til prosessen gjør at notasjonen i seg selv ikke krever vanskelige mentale operasjoner. På den andre siden vil det være vanskelig å forstå funksjonaliteten til applikasjonen ved å se på informasjonsradiatoren og brukerhistoriekortene. Dersom kortene er vanskelig å forstå, eller krever en spesifikk kompetanse, må meningen forklares gjennom samtale. Videre er det skjulte avhengigheter mellom kort, den implisitte koblingen mellom prosess og applikasjonsperspektiv, samt abstraksjonen til brukerhistorier krever vanskelige mentale operasjoner.

Midlertidighet: Fordi kortene er håndskrevne, fleksible og tynne⁷ fysiske kort, uttrykker de en viss grad av midlertidighet. Midlertidigheten kan også uttrykkes direkte i

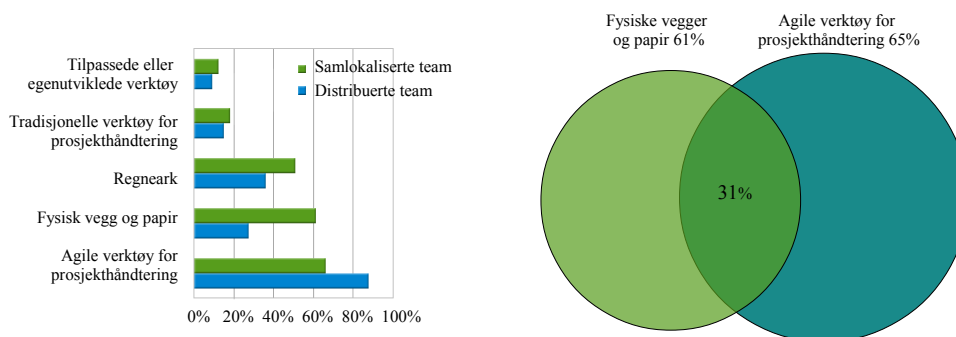
⁷Eng.: flimsiness

kortet, eksempelvis ved måten kortene er skrevet. Informasjonsradiatoren støtter også midlertidighet, gjennom måten kortene kan bli sortert. Merkelapper kan bli lagt til eller slettet og kort kan midlertidig fjernes.

Progressiv evaluering: Brukerhistoriekortene støtter ikke progressiv evaluering. Progressiv evaluering kan oppnås gjennom rask tilbakemelding og fungerende kode. Notasjonen på kortene uttrykker hvor langt brukerhistorien eller oppgaven har kommet i prosessen, men ikke noe om hvor «riktige» de er i forhold til applikasjonen.

2.1.6 Digitale verktøy for prosjekthåndtering

En spørreundersøkelse gjennomført av Azizyan et al. (2011) identifiserer hvilke verktøy som brukes i det smidige programvaremiljøet, og hvilket behov disse verktøyene bør dekke. Svarteltakelsen inkluderer 121 svar fra 120 ulike selskap, fordelt på 35 nasjoner. Azizyan et al. (2011) deler verktøy inn i fem kategorier: smidige verktøy for prosjekthåndtering, fysisk informasjonsradiator og papir, regneark, tradisjonelle verktøy for prosjekthåndtering og tilpassede eller egenutviklede verktøy. Figur 3(a) viser fordelingen av kategoriene, gruppert i samlokaliserte og distribuerte team. Et resultat fra denne undersøkelsen viser at 61% av samlokaliserte team bruker smidige verktøy for prosjekthåndtering, 65% bruker fysiske verktøy og 31% bruker begge deler, som illustrert i figur 3(b).

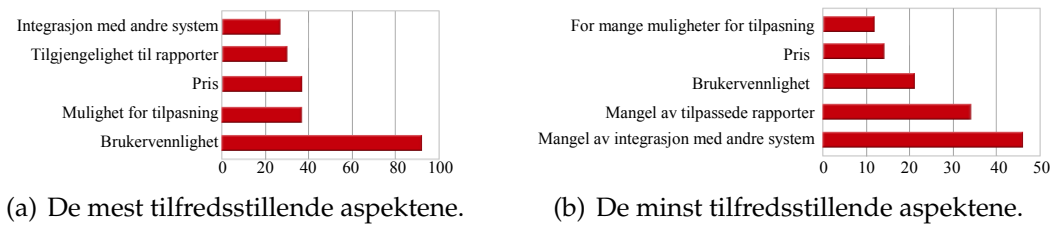


(a) Kategorier av verktøy som blir brukt i samlokaliserte eller distribuerte team.

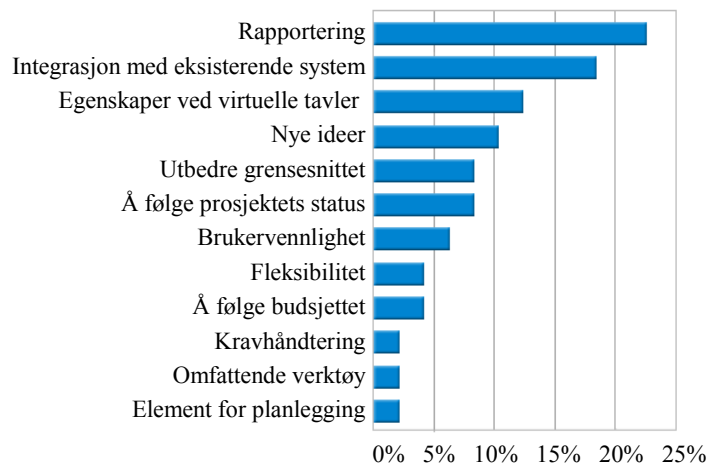
(b) Overlappet mellom fysiske og smidige verktøy for prosjekthåndtering som brukes av samlokaliserte team.

Figur 3: viser kategorisering bruk av verktøy blant informantene i (Azizyan et al., 2011).

Undersøkelsen til Azizyan et al. (2011) identifiserer også de mest og minst tilfredsstillende aspektene med verktøyene som ble brukt av informantene, som vist i figur 4(a) og 4(b). Det mest tilfredsstillende aspektene var brukervennlighet og mulighet for tilpasning. De minst tilfredsstillende aspektene var manglende integrasjon med andre system og tilpassede rapporter. 51 informanter i undersøkelsen kommenterte hvilke egenskaper verktøyene bør ha. Disse egenskapene vises i figur 5, og viser at rapportering og integrasjon var de egenskapene som informantene hadde størst behov for.



Figur 4: De mest og minst tilfredsstillende aspektene til verktøyene som ble brukt av informantene i (Azizyan et al., 2011).



Figur 5: viser grupperte kommentarer som påpekte hvilke egenskaper som behøves i verktøy, presentert i prosent (Azizyan et al., 2011).

Kurpicz (2011) har utført en undersøkelse hvor det kom fram at flere av informantene hevdet at det ikke er behov for et datastøttet verktøy i Scrum. De hevdet at hovedfordelen med Scrum er den menneskelige faktoren, noe man ikke oppnår med digitale verktøy. Kurpicz (2011) gir et eksempel hvor en utvikler går mot informasjonsradiatoren og nøler

når han skal velge en oppgave. Denne aktiviteten kan oppfattes av de andre i teamet og kan føre til kommunikasjon innad i teamet. Sharp & Robinson (2008) gjør også et poeng av at det å forlate arbeidsplassen og interagere med informasjonsradiatoren er en aktivitet som ubevisst øker teamets forståelse både for aktiviteten og prosessen. Sharp & Robinson (2008) trekker frem en rekke studier som viser at det fysiske mediet påvirker koordinering og samarbeid ved en rekke aktiviteter, og konkluderer med at en nøkkelkonsekvens med den fysiske naturen til artefaktene er at aktiviteten ikke enkelt kan skjules. I tillegg hevder Sharp & Robinson (2008) at man må være disiplinert i bruken av informasjonsradiatoren og at selve aktiviteten med informasjonsradiatoren gir et signal om at programmerere tar ansvar for oppgavene de interagerer med. En selvmotsigelse som ble avdekket av Kurpicz (2011) var at noen av informantene brukte datastøttede verktøy selv om de hevdet at det ikke fantes et slikt behov. De mest populære årsakene til å bruke datastøttede verktøy var: rapportering, dokumentasjon, teamet var distribuert, for å brukes i en ikke-smidig kontekst, juridiske årsaker eller ved samarbeid med andre team. Verktøyene ble brukt i tillegg til å bruke en fysisk informasjonsradiator (Kurpicz, 2011).

Flere i miljøet rundt XP anbefaler å bruke fysiske kort framfor programvare (Beck & Andres, 2004). Cohn (2004) trekker frem at en av fordelene med fysiske kort er at de er lavteknologiske av natur, som er en konstant påminnelse om at brukerhistorier er ufullstendige. Ved å bruke en programvare legges det gjerne til unødvendige detaljer til brukerhistoriene (Cohn, 2004). Videre hevder Cohn (2004) at en ulempe med den fysiske begrensningen til kortene er at det ikke er nok plass til å skrive tilstrekkelige akseptansetester på kortene. Kortene kan mistes eller bli ødelagt, og kan ikke enkelt bli delt med andre (Sharp et al., 2009). Brukerhistorier som er lagt inn i en programvare kan enklere sorteres, og kan tildeles utviklere eller kunder som ikke befinner seg på samme sted. Cohn (2004) anbefaler derfor å begynne med fysiske kort og vurdere hvordan dette fungerer, og heller bytte til en programvare hvis dette er hensiktsmessig.

Dubakov & Stevens (2008) diskuterer ulike fremgangsmåter for å lede smidige prosjekter. De utfører en sammenligning av verktøy som fysiske brukerhistoriekort, whiteboard, regneark, tradisjonelle prosjekthåndteringsverktøy og smidige prosjekthåndteringsverktøy. I artikkelen kommer det frem en rekke fordeler og ulemper ved de ulike verktøyene, presentert i tabell 2. Ut fra sammenligningen predikerer Dubakov & Stevens (2008) tre punkter for fremtidige smidige verktøy:

1. Det vil oppstå en kobling mellom fysiske og digitale verktøy.
2. Systemer for smidig prosjekthåndtering utvikler seg til løsninger som integrerer hele livssyklusen ved utvikling.
3. Flere virksomheter vil bruke distribuerte team, og smidige verktøy vil i større grad fokusere på å støtte samarbeid i slike team.

Verktøy	Fordeler	Ulemper
Kartotekkort og store synlige diagram	<ul style="list-style-type: none"> • Lett å lære og lett å bruke • Fleksible • Billige 	<ul style="list-style-type: none"> • Fungerer ikke for distribuerte team • Fungerer ikke for store team • Vanskeliggjør rapportering • Må manuelt oppdateres
Generelle programvareverktøy, f.eks. wiki, regneark	<ul style="list-style-type: none"> • Fungerer for distribuerte team • Billige 	<ul style="list-style-type: none"> • Fungerer ikke for store team • Begrenset rapportering • Begrenset synlighet • Må manuelt oppdatere tidsbruk
Tradisjonelle prosjekthåndteringsverktøy	<ul style="list-style-type: none"> • Eksisterer kanskje i virksomheten fra tidligere • Støtte for å fordele personalressurser 	<ul style="list-style-type: none"> • Støtter ikke smidige utviklingskonsepter • Begrenset rapportering • Begrenset synlighet
Smidig prosjekthåndteringsverktøy	<ul style="list-style-type: none"> • Støtter distribuerte team • Støtter store team • Sanntidsrapportering • Integrerte løsninger, f.eks. kobling mot kildehandtering og bughåndteringssystemer 	<ul style="list-style-type: none"> • Begrenset synlighet • Signifikant læringskurve • Kan være kostbare

Tabell 2: Tabellen viser identifiserte fordeler og ulemper ved ulike verktøy (Dubakov & Stevens, 2008)

2.2 Videovegg

I stedet for begrepet videovegg bruker Andrews, Endert, Yost & North (2011) begrepet «stor, høyoppløselig skjerm», som kan i følge dem defineres på flere måter. Begrepet kan defineres i forhold til teknologien som brukes, altså skjermmiljøer som er større enn tradisjonelle skjermer. Andrews et al. (2011) påpeker at denne definisjonen er subjektiv, og at det er ens egen oppfattelse av skjermen som definerer den. Videre skrives det at når teknologien blir bedre og tradisjonelle skjermer endres vil skjermer som en gang var store og høyoppløselige ikke lenger inkluderes under denne definisjonen. Andrews et al. (2011) nevner derfor en annen definisjon, hvor begrepet beskrives i forhold til mengden data som visuelt kan representeres, samt muligheten til å representere data i ulike former, i ulik skala og enheter (f.eks. dokumenter og nettsider). For å bruke denne definisjonen må man definere hva som er en dataenhet, noe som varierer mellom applikasjoner og datasett. Andrews et al. (2011) skriver derfor at selv om disse to definisjonene kan være nyttige, ønsker de å benytte en tredje definisjon. Store, høyoppløselige skjermer defineres som skjermmiljøer i menneskelig skala. Med menneskelig skala menes det at skjermens størrelse og oppløsning er nært relatert til menneskets synsfelt og rekkevidde. Med skjermmiljøer som går under denne definisjonen kan man bytte ut virtuell med fysisk navigering (snu, lene og bevege seg rundt), og således tillate brukere å utnytte menneskelige egenskaper som romforståelse, proprioepsjon og spatialt minne. Å bytte ut virtuell med fysisk navigering er viktig fordi det varsler om en endring i brukernes oppførsel. Fysisk navigering krever at det tas designhensyn basert på menneskelige evner, fremfor de teknologiske begrensningene til mediet (Andrews et al., 2011).

Behovet for fysisk navigasjon viser seg tydelig i systemene HyperWall og HyperWall2, utviklet av NASA⁸, hvor skjermflaten i førstnevnte er fem kvadratmeter, fordelt på 49 skjermer og i sistnevnte 21 kvadratmeter fordelt på 128 skjermer (Sandstrom, Henze & Levit, 2003; Mewhinney & Fay, 2008). Videre er det identifisert av Ball, North & Bowman (2007) at fysisk navigasjon er overlegen i forhold til virtuell navigasjon ved utførelse av virtualiseringsoppgaver av geografisk informasjon.

Ni, Schmidt, Staadt, Livingston, Ball & May (2006) beskriver bruken av videovegger som utbredt og variert. Dette illustrerer de ved å presentere en rekke eksempler, deriblant systemer som støtter vitenskapelige visualiseringer, utdanning og trening innen astronomi,

⁸NASA: National Aeronautics and Space Administration

bioinformatikk og geografi, samt kommando- og kontrollsystemer innenfor militæret, romfart og telekommunikasjon. Generelt for disse systemene er at de har muligheten til å vise store mengder informasjon, hvilket kan støtte samarbeid og kommunikasjon (Ni et al., 2006; Andrews et al., 2011). Videovegger er i følge Ebert, Thelen, Olech, Meyer & Hagen (2010) forutbestemt for å støtte samarbeid. Ebert et al. (2010) skriver at videovegger kan være et effektivt verktøy under diskusjoner ved planlegging og utvikling, siden mediet tilrettelegger for interaksjon mellom flere brukere som er plassert foran skjermen.

Bi & Balakrishnan (2009) har utført en studie hvor de ønsket å se hvordan videovegger kan brukes som skrivebordsmiljø, og identifiserer i denne sammenheng en rekke fordeler: oppgaver som krevde flere applikasjoner ble enklere å utføre, da man hadde god oversikt over alle applikasjonene samtidig. Det ble enklere å bruke informasjonsrike applikasjoner som regneark på grunn av mindre behov for navigasjon, da et større område i applikasjonene var synlig. Perifere applikasjoner, som epostklient og kalender, kan alltid være synlig, og dermed øke produktiviteten ved at man ikke trenger å bruke tid på å sjekke disse regelmessig.

Selv om en rekke bruksområder og flere fordeler ved videovegger er identifisert, er det flere begrensninger og praktiske hindringer som må betraktes når man utvikler applikasjoner for slike system. Robertson, Czerwinski, Baudisch, Meyers, Robbins, Smith & Tan (2005) og Ni et al. (2006) skriver blant annet at etter hvert som den totale skjermstørrelsen øker, vil det bli vanskeligere å holde rede på hvor markøren befinner seg. Ni et al. (2006) skriver også om problemer med å nå elementer som er spredt utover skjermflaten, dette eksemplifiseres ved at det er svært tidkrevende å flytte et ikon fra nedre høyre skjerm til søppelkurven på venstre skjerm. Videre poengterer Robertson et al. (2005) at det er generelt dårlig støtte for konfigurering av skjermoppsettet på videovegger, og at dette er unødvendig tungvint.

Teknologisk kan videovegger settes opp svært forskjellig, og man skiller mellom LCD⁹-baserte og projektorbaserte systemer (Ni et al., 2006). HyperWall og HyperWall2 er LCD-baserte, men det finnes også projektorbaserte systemer. I hovedsak skriver Ni et al. (2006) at LCD-baserte systemer er billigere og har lengre levetid, mens projektorbaserte systemer har fordelen ved å gi et sømløst bilde. På tross av at LCD-baserte systemer har rammer rundt hver skjerm har Ball & North (2005) observert at brukere ofte benytter seg av disse rammene for å skille mellom ulike oppgaver og ideer. Likevel ansees rammene som et

⁹LCD: Liquid Crystal Display

problem når informasjon vises over flere skjermer samtidig, og i en studie av Ebert et al. (2010) ble det forsøkt å benytte projektor i tillegg til LCD-skjermer. Projektorene ble brukt til å vise bilde på rammene, slik at rammene ble en integrert del av skjermbildet.

Robertson et al. (2005) har forsøkt å løse noen av begrensningene ved videovegger, deriblant foreslås det teknikker for å identifisere hvor markøren befinner seg. I innstillinger for musepeker under Microsoft Windows kan man slå på markørgjenfinning, markøren vil da signalisere hvor den befinner seg hver gang Ctrl-knappen trykkes. Robertson et al. (2005) videreutvikler Microsoft sin løsning slik at den er enda mer passende for videovegger. De utvikler ytterligere to funksjoner for å bedre interaksjon med mus, hvor den ene gjør det mulig å navigere lengre distanser ved å la markøren fortsette i en definert retning til brukeren gjenopptar kontroll, og den andre forbedrer måten man bytter mellom åpne applikasjoner.

2.3 Interaksjon med videovegger

Tradisjonell interaksjon med mus og tastatur er i følge Ni et al. (2006) ikke tilstrekkelig som interaksjonsmetode på videovegger. Ni et al. (2006) skriver i sin litteraturgjennomgang på interaksjonsmetoder for videovegger at flere interaksjonsmetoder er blitt undersøkt, inkludert naturlige gester, stemmegjennkjenning, interaksjon med flere hender og måter å forbedre rekkevidden til brukerne. Selv om disse interaksjonsmetodene virker lovende, må de evalueres i forhold til spesifikke oppgaver for bedre å forstå hvor effektive de er med tanke på interaksjon med videovegger (Ni et al., 2006).

Miller, Robinson, Wang, Chung & Quek (2008) utfører en evaluering på en horisontal videovegg, hvor det fokuseres på interaksjon med PDA¹⁰, symboler, berøringsgester og stemmegjennkjenning. Evalueringen identifiserer styrker og svakheter med alle interaksjonsmetodene. Deriblant identifiseres interaksjon med PDA som allsidig med tanke på enkle handlinger, som velging, slipping og lenking av elementer, men også ved oppgaver som tekstredigering. Interaksjon med symboler beskrives som en lettvekts, robust og enkelt konfigurerbar interaksjonsmetode som gir god bevegelseskontroll. Gestbasert interaksjon gir brukeren kontroll over systemet uten å være avhengig av andre interaksjonanordninger som PDA eller symboler. Det eneste unntaket er når man skal redigere tekst. Stemmegjennkjenning gir brukeren mulighet til å bruke en naturlig

¹⁰PDA: Personal digital assistant

kommunikasjon, som frigjør andre ressurser til multitasking.

Malik, Ranjan & Balakrishnan (2005) har gjennom sitt arbeid utviklet en portabel berøringsflate som kan benyttes for å styre videovegger. Denne berøringsflaten har muligheten til å kjenne igjen håndbevegelser. Håndbevegelsene gjenkjennes av systemet, og kan brukes for å utføre operasjoner. Videre krever interaksjonsmetoden at brukerne sitter ved et bord.

Cao & Balakrishnan (2003) har utviklet en interaksjonsmetode hvor man benytter en tryllestav som kontinuerlig spores i tre dimensjoner. Ved hjelp av to webkamera som sporer tryllestaven, samt en rekke gester, kan man utføre operasjoner som å peke, velge, tilte, rotere, skubbe eller trekke elementer. Cao & Balakrishnan (2003) hevder at interaksjonsmetoden er lett å bruke og enkel å forstå, spesielt hvis man begrenser tryllestavens gester og positurer.

Shoemaker, Tsukitani, Kitamura & Booth (2010) utførte en studie hvor de evaluerte interaksjonsmetoder som er sentrert rundt kroppen til brukeren. Som en del av studien er det utviklet programvare som, ved hjelp av enten et magnetfelt eller markerte kroppsledd, kan skape en tredimensjonal modell av menneskekroppen. En skyggerepresentasjon av modellen vises på videoveggen, hvor brukerens personlige menyvalg plasseres. Mer spesifikt kan implementasjonen til Shoemaker et al. (2010) håndtere panorering og zooming, samt frihåndstegning og skrijving via et virtuelt tastatur. Hver bruker har en Nintendo Wii kontrollenhet ¹¹ i hver hånd for å kunne lage klikkeventer. Selv om studien viste at alle brukerne forstod konseptet bak skyggerepresentasjonen og interaksjonsmetoden, ble det identifisert elementer som kan forbedres. Blant elementene var bruken av Nintendo Wii kontrollenheten. Enheten er designet, og ble brukt av deltakerne som en pekeenhet, men var i utgangspunktet kun tiltenkt som en fjernkontroll.

Shoemaker et al. (2010) nevner i videre arbeid at Microsoft Kinect¹² kan være aktuell å implementere istedet for deres egenutviklede kameraløsning. En slik implementasjon beskrives av Gallo, Placitelli & Ciampi (2011), hvor de benytter Kinect sammen med et tradisjonelt projektorsystem for å utforske medisinsk bildemateriell.

¹¹Nintendo Wii: en spillkonsoll utviklet av Nintendo og Wii Remote, forkortet Wiimote, er en kontrollenhet til spillkonsollen. Teknologien beskrives i seksjon 5.2.3.

¹²Kinect: en bevegelsessensor utviklet av Microsoft for spillkonsollen Xbox 360. Teknologien beskrives i seksjon 5.2.2.

Som hos Shoemaker et al. (2010) brukes hånd- og armgester for å utføre elementære oppgaver som peking, velging, zooming, panorering, rotering, samt mer komplekse oppgaver som å manuelt velge og hente ut interesseområder. Gallo et al. (2011) beskriver teknologien i Kinect som veldig god, og det henvises til at bruk av slik teknologi kan redusere forsinkelser i operasjoner med opptil to timer. En annen fordel med teknologien i Kinect er at den fungerer i ulike lysforhold, hvilket kan forekomme i operasjonssaler. Videre argumenteres det for bruk av Kinect, da man ikke ønsker å introdusere enheter som ikke er sterile i en operasjonssal.

2.4 Oppsummering

I dette kapitlet ble relevant teori og tidligere forskning innenfor tre områder beskrevet. Innenfor smidige artefakter ble teori om informasjonsradiatorer, burndowndiagrammer, brukerhistorie- og oppgavekort presentert. I tillegg ble studien til Sharp, Robinson og Petre, som var utgangspunktet for denne masteroppgaven, presentert. Avslutningsvis i seksjonen om smidige artefakter ble tre studier som vurderer digitale artefakter i forhold til fysiske artefakter gjengitt for å gi et innblikk i behovet som fremmes av det smidige miljøet. Deretter fulgte en presentasjon av teori og tidligere forskning innenfor videovegger og interaksjon med store høyoppløselige skjermer. I seksjonen om videovegger ble det presentert fordeler og utfordringer, typer og bruksområder samt interaksjonsmetoder som er funksjonelle til bruk med videovegger.

3 Metode

Med bakgrunn i problemstillingen, presentert i kapittel 1, redegjøres det her for aktuelle metoder som kan belyse denne. Kapitlet vil innledningsvis beskrive designforskning og retningslinjer innenfor dette paradigmet. Videre presenteres to evalueringsformer: brukbarhetstesting og brukbarhetsinspeksjon. Hvordan denne studien benytter disse metodene presenteres i forskningsdesignet, kapittel 4.

3.1 Designforskning

Hevner, March, Park & Ram (2004) skiller mellom to hovedparadigmer: designforskning og atferdsforskning¹³. I følge Hevner & Chatterjee (2010) handler designforskning om å skape innovative artefakter i den hensikt å svare på spørsmål som er relevante i forhold til menneskelige problem, og dermed bidra til ny kunnskap. Atferdsforskning starter med en hypotese, deretter samler forskeren inn data med hensikt å bekrefte eller avkrefte hypotesen. Atferdsforskning fokuserer på å forklare menneskelig eller organisatorisk oppførsel, mens designforskning handler om å tilegne seg kunnskap gjennom å skape og evaluere IT-artefakter. Hevner et al. (2004) beskriver syv retningslinjer for designforskning, disse er:

1. Design som et artefakt: Designforskning må produsere et artefakt som er realiserbart, i form av et sett med begreper, en modell, metode eller eksempel.
2. Problemrelevanse: Formålet ved designforskning er å utvikle teknologibaserte løsninger til viktige og relevante forretningsproblem.
3. Designevaluering: Bruken, kvaliteten og virkningen må være grundig demonstrert via velutførte metoder for evaluering.
4. Forskningsbidrag: For at designforskning skal være hensiktsmessig må den gi klare bidrag som kan verifiseres innenfor kunnskapsdomenet til designartefaktet, designfundamentet, og/eller designmetodologier.
5. Nøyaktig forskning: Designforskning baserer seg på bruk av rigide metoder i både skapelsen og evalueringen av artefaktet.
6. Design som en søkeprosess: Søken for en hensiktsmessig artefakt krever at man

¹³Eng.: behavioural science

bruker tilgjengelige midler for å nå et ønsket mål, samtidig som man tar hensyn til begrensningene i problemområdet.

7. Kommunikasjon av forskningen: Designforskning må bli presentert på en hensiktsmessig måte, både til et teknologi- og forretningsorientert publikum.

Innenfor designforskning er evaluering og testing av artefaktet sentrale aktiviteter. Feltstudier av artefaktet i designforskning utføres som en del av relevanssyklusen for å identifisere om flere iterasjoner i denne syklusen er nødvendig, eller om artefaktet kan sies å være ferdig (Hevner, 2007). Før en slik studie kan utføres er det den interne designsyklusen som er sentral. Denne syklusen, med utvikling, evaluering og tilbakemeldinger, er hjertet i en hver studie innenfor designforskning (Hevner, 2007). Videre fremgår det i Hevner (2007) at evalueringene i denne syklusen bør utføres grundig og nøyaktig med vitenskapelige metoder.

Sharp et al. (2011) skiller evalueringsmetoder innen interaksjonsdesign i tre brede kategorier:

1. Kontrollert miljø med brukere, eksempelvis brukertesting/brukbarhetstesting.
2. Naturlig miljø med brukere, eksempelvis feltstudier.
3. Alle andre miljø uten brukere, eksempelvis brukbarhetsinspeksjon.

Blant disse tre kategoriene er hovedmålet ved brukbarhetstesting å avgjøre om et grensesnitt blir forstått av målgruppen til systemet, slik at grensesnittet kan brukes slik det var designet (Sharp et al., 2011). Denne formen for evaluering utføres som regel i kontrollerte miljøer, som et laboratorium, hvor man ikke blir forstyrret eller avbrutt og hvor man samtidig har større kontroll over testen i sin helhet. På den andre siden er feltstudier som utføres i et naturlig miljø, med mulighet til å vurdere hvordan et produkt benyttes når man ikke lenger er skjermet fra omverden.

3.2 Brukertesting

Brukertesting foregår som regel i kontrollerte miljøer, med et mål om å identifisere hvorvidt produktet som utvikles kan brukes av målgruppen til produktet (Sharp et al., 2011). I følge Sharp, Rogers & Preece kan brukbarhet måles med en kombinasjon av metoder, deriblant intervju og observasjon.

3.2.1 Intervju

Et intervju ses på som en *samtale med et mål*, og kan struktureres på ulike måter (Sharp et al., 2011). På den ene siden er ustrukturerte intervju, en prosess hvor intervjuet er utforskende rundt et spesifikt emne og ofte går i dybden på temaet. På den andre siden er strukturerte intervju, hvor spørsmålene som stilles ofte er lukket, definert på forhånd og like for alle som intervjues. Et semistrukturert intervju er en kombinasjon av disse to teknikkene som kan ha både åpne og lukkede spørsmål. Semistrukturerte intervju benytter i likhet med strukturerte intervju en intervjuguide, men gir deltakeren større fleksibilitet i forhold til hvordan spørsmålene besvares (Bryman, 2008).

3.2.2 Think-aloud

Think-aloud er en teknikk som kan brukes under observasjon, hvor man ber deltakerne om konstant å verbalisere sine tanker mens de utfører et sett med oppgaver (van den Haak, De Jong & Jan Schellens, 2003). Teknikken er benyttet siden 1980-tallet innen HCI og regnes som veletablert (van Someren, Barnard & Sandberg, 1994; Nielsen, Clemmensen & Yssing, 2002). En av fordelene ved think-aloud er at tankene rundt elementene og handlingene er i korttidsminnet, mens en risikerer at tankene blir glemt ved et retrospektivt studium (Nielsen et al., 2002).

Ericsson og Simon presenterte i 1984 den klassiske og mest siterte teknikken for think-aloud innenfor HCI, men det er et stort gap mellom den klassiske teknikken og hvordan think-aloud benyttes i praksis (Boren & Ramey, 2000). Boren & Ramey (2000) savner en detaljert beskrivelse av hvordan think-aloud skal utføres og etterlyser teoretisk motiverte retningslinjer for utførelsen. De foreslår et alternativt rammeverk basert på verbal kommunikasjon, hvor fokuset er å bedre oppnå Ericsson og Simons mål. Boren og Ramey har derfor utviklet retningslinjer i forhold til definerte rammer, språkbruk, oppførsel og for å få frem ytterligere informasjon.

Definerte rammer: Retningslinjen beskriver at det må skapes en atmosfære hvor deltakerne forstår at det er produktet som blir testet, ikke dem selv. Deltakeren bør også gjøres oppmerksom på at han er en viktig bidragsyter, eller på annen måte verdifull for prosessen. Samtidig er det vesentlig at deltakeren forstår at observatørens rolle primært er å lytte og lære av deltakeren. Videre fastslås det

at teknisk utstyr, eksempelvis et videokamera, skal plasseres diskret for å ikke virke forstyrrende på deltakeren.

Språkbruk: Konstant bekreftelse i form av korte tilbakemeldinger som «Mm hm» vil oppfordre deltakeren til å fortsette å verbalisere sine tanker.

Oppførsel: Retningslinjene til Boren & Ramey (2000) tillater interaksjon mellom observatør og deltaker når det er nødvendig. Interaksjon kan være aktuelt i tilfeller hvor feil i artefaktet må korrigeres, og hvor deltakeren utfører handlinger som er utenfor interesseområdet eller ikke er relevant for studien.

Få fram ytterligere informasjon: Observatør kan gripe inn eller stille oppfølgingsspørsmål for å få klarhet i hva deltakeren mener, men dette bør begrenses til et minimum da resultatet kan være at deltakeren blir avsporet fra oppgaven.

Krahmer & Ummelen (2004) har sammenlignet Ericsson og Simons med Boren og Rameys fremgangsmåte ved å utføre brukertester med think-aloudteknikkene på en nettside. Studien viste at begge teknikkene identifiserte likt antall brukbarhetsproblemer, men deltakerne i brukertestene som brukte Boren og Rameys teknikk var mindre forvirret og klarte samtidig å utføre flere oppgaver.

Think-aloudteknikken gjør det mulig å samle vesentlige mengder kvalitative data fra flere deltakere. Disse datasamlingene betegnes som protokoller, og er analyseenheter som brukes for å forstå hva deltakere tenker og hvordan de tolker ulike elementer. Protokollene kan analyseres på ulike nivå, hvor et overordnet nivå fokuserer på emner og temaer, og på detaljnivå vurderer hvert ord, uttalelse og gest (Sharp et al., 2011). Sharp et al. (2011) skriver videre at uansett hvilket detaljnivå man ønsker å analysere, må innholdet kategoriseres i forhold til et skjema. Kategoriskjemaet er avhengig av målet ved studien og kan opprettes i forhold til innholdet i protokollene, være forhåndsdefinert eller opprettes samtidig som datamaterialet kategoriseres. Sharp et al. (2011) hevder at en utfordring ved kategoriskjemaet er at kategoriene ikke må være overlappende, samt ha et passende detaljnivå i forhold til målet ved analysen. Ved å la flere personer kategorisere samme data kan man demonstrere hvor nøyaktig kategoriskjemaet er, hvilket kan si noe om reliabiliteten ved analysen. Hvis svært ulike resultat er oppnådd av to personer er enten opplæring eller kategoriskjemaet for dårlig (Sharp et al., 2011).

3.2.3 Deltakere

Målet ved datainnsamlingen avgjør hvilke deltakere man inkluderer (Sharp et al., 2011). Tilstrekkelig antall deltakere i en brukerteste er fire til fem deltakere i følge Nielsen (1994a). Han argumenterer for at mindre og mindre brukbarhetsproblemer identifiseres for hver deltaker, og ved fem deltakere vil 75% av alle brukbarhetsproblemer være identifisert.

3.3 Brukbarhetsinspeksjon

Brukbarhetsinspeksjon er en samlebetegnelse for metoder som baserer seg på å la personer evaluere, inspisere eller undersøke brukbarhetsrelaterte aspekt ved et brukergrensesnitt. En definerende karakteristikk ved brukbarhetsinspeksjon er at evalueringen av spesifikke elementer i et brukergrensesnitt er avhengige av vurderingene fra dem som utfører inspeksjonen (Nielsen & Mack, 1994).

Nielsen & Mack (1994) skiller mellom metoder for å evaluere brukergrensesnitt i fire grunnleggende kategorier:

Automatiske: Måling av brukbarhet ved å kjøre en spesifikasjon gjennom en programvare som er spesifikt laget for å evaluere brukergrensesnitt.

Empiriske: Testing av brukergrensesnittet mot reelle brukere.

Formelle: Måling av brukbarhet ved å bruke nøyaktige modeller og formler.

Uformelle: Metoder som benytter tommelfingerregler, den generelle kunnskapen og erfaringen til personen som utfører evalueringen.

Uformelle metoder baserer seg på å forstå brukeren gjennom kunnskap kodet i heuristikker, data som er samlet utenfra eller modeller som forutser brukerens oppførsel (Sharp et al., 2011). Nielsen & Mack (1994) kategoriserer brukbarhetsinspeksjon under uformelle metoder. En brukbarhetsinspeksjon har vanligvis mål om å finne brukbarhetsproblemer i et eksisterende design, for så å bruke problemene for å gi anbefalinger til hvordan en kan forbedre brukbarheten til designet (Nielsen & Mack, 1994). Inspeksjonen kan utføres uten at brukere er tilstede, noe som kan være fordelaktig når brukere ikke er tilgjengelige, eller det å involvere dem er kostbart og/eller tidkrevende (Sharp et al., 2011).

Nielsen & Mack (1994) hevder også at enkelte studier konkluderer med at flere brukbarhetsproblem blir oversett under brukbarhetsinspeksjon, men oppdaget under empirisk testing, og vice versa. Dette indikerer at de beste resultatene vil komme ved å kombinere empiriske tester med brukbarhetsinspeksjoner (Nielsen & Mack, 1994).

Brukbarhetsinspeksjon er som tidligere påpekt et generisk begrep for flere metoder, og følgende avsnitt vil presentere to forskjellige metoder: heuristisk evaluering og kognitive dimensjoner.

3.3.1 Heuristisk evaluering

Heuristisk evaluering er en metode for å identifisere brukbarhetsproblem i et brukergrensesnitt (Nielsen, 1994b). Heuristisk evaluering involverer en liten gruppe med eksperter som evaluerer grensesnittet og ser hvordan det stemmer overens med anerkjente prinsipper for brukervennlighet, kalt heuristikker (Nielsen, 1994b). Nielsen og hans kollegaer identifiserte en liste med heuristikker som tok utgangspunkt i en analyse av 249 brukbarhetsproblem (Nielsen, 1994c), en liste som i ettertid har blitt revidert¹⁴ (Sharp et al., 2011). En heuristisk evaluering resulterer i en rekke identifiserte brukbarhetsproblem (Nielsen, 1994b).

Heuristisk evaluering har ifølge Nielsen (1994b) fordelene av å være lett å lære, samt hurtig og billig å utføre. Metoden er også særs effektiv til å finne brukbarhetsproblem (Jeffries, Miller, Wharton & Uyeda, 1991; Desurvire, 1994), men er kun gyldig når evalueringen er gjennomført av flere kvalifiserte eksperter (Jeffries & Desurvire, 1992). En svakhet ved metoden er at en kan overse alvorlige brukbarhetsproblem (Jeffries & Desurvire, 1992; Karat, 1994) og at metoden kan gi falske positive (Sharp et al., 2011).

I eksperimenter utført av Jeffries et al. (1991) og Desurvire (1994) ble det vist at brukbarhetsinspeksjon fungerer svært bra, da eksperter som brukte heuristisk evaluering fant flere problemer enn ved brukbarhetstesting av samme system. I følge Jeffries & Desurvire (1992) ble fordelene ved heuristisk evaluering tolket av flere som «bevis» for at brukbarhetstesting var sløsing med tid, hvilket ikke var tilfelle. Jeffries & Desurvire (1992) reagerte på denne misoppfattelsen, og publiserte derfor en artikkel der de kritiserte tolkningen av resultatet for å være en kraftig en overdrivelse.

I samme artikkel påpeker Jeffries & Desurvire (1992) flere begrensninger ved heuristisk

¹⁴Nielsens ti heuristikker: http://www.useit.com/papers/heuristic/heuristic_list.html

evaluering i forhold til brukbarhetstesting. Metoden krever trente og kvalifiserte personer for å utføre evalueringen samt tre til fem personer for at den skal være effektiv. Videre skriver Jeffries & Desurvire (1992) at heuristisk evaluering bare identifiserte halvparten av de samme brukbarhetsproblemene som kom frem under brukbarhetstesting på samme artefakt. Brukbarhetsproblemene fra brukbarhetstesting ble også vurdert som mer alvorlige. Flere av brukbarhetsproblemene fra heuristisk evaluering ble vurdert som mindre viktige, og ofte var det snakk om «smak og behag» fra de som evaluerte brukergrensesnittet. Jeffries & Desurvire (1992) skriver at ulike HCI-eksperter kan være uenige om validiteten og verdien til funnene som avdekkes med heuristisk evaluering.

3.3.2 Kognitivt dimensjonsrammeverk

Green (1989) introduserer kognitive dimensjoner som et rammeverk for å analysere brukbarhet fra et overordnet syn. Mer spesifikt er rammeverket ment som et generelt verktøy for å støtte utviklere og andre som ikke er HCI-eksperter ved evaluering av brukbarhet i informasjonssystemer (Green & Blackwell, 1998). Green & Hevner (2000) beskriver rammeverket som et sett med diskusjonsverktøy som kan bli brukt av designere og personer som evaluerer design. Rammeverket beskrives som komplimentært til andre evalueringmetoder, deriblant heuristisk evaluering. Analysen betegnes av Green & Blackwell (1998) blant annet som rask å lære, rask å anvende og mulig å utføre på et hvilket som helst stadiet av utviklingsprosessen. Analysen av kognitive dimensjoner gir *ikke* informasjon om det estetiske og følelsesmessige aspektet av brukervennlighet, men fokuserer heller på å forbedre spesifikke brukbarhetsproblemer (Green & Blackwell, 1998). Rammeverket er tiltenkt som analyseverktøy for informasjonsartefakter, representasjoner og notasjonelle systemer generelt (Green, 1991).

Blackwell (2000a) hevder at ethvert system er en kombinasjon av notasjon og miljø, hvor notasjon er symbolene som en bruker ser eller manipulerer, og brukervennlighet er en funksjon av denne notasjonen og dens miljø. Deler av notasjonen kan være separat fra resten av systemet, og kalles da subenheter (Blackwell, 2000a). Subenheter deles inn i hjelpeenheter og enheter som redefinerer notasjonen. Et eksempel på en hjelpeenhet er merkelapper som henges på skjermen for å fungere som påminnelser, mens det å definere en hurtigtast for å utføre en operasjon er et eksempel på en enhet som redefinerer notasjonen. Slike subenheter har ofte en egen notasjon som er skilt fra notasjonen til hovedsystemet, og har et uavhengig sett med kognitive dimensjoner (Blackwell, 2000a).

De kognitive dimensjonene i rammeverket er et definert vokabular. Hvor dimensjonene representerer vanlige faktorer for brukervennlighet som erfarne designere kanskje har merket seg, men ikke har et navn for. Rammeverket forsøker i så måte å gi navn på disse faktorene, slik at diskusjoner rundt disse blir enklere. Videre er rammeverket basert på observasjoner om at det ikke finnes et perfekt brukergrensesnitt, og at et hvert grensesnitt har ulike kompromiss. Hvilket gjenspeiles i dimensjonene, da det ikke er mulig å lage et design som er perfekt i forhold til hver dimensjon. Blackwell (2000a) skriver at forbedringer i én dimensjon ofte resulterer i svekkelser i en annen. En sammenfattet beskrivelse av de kognitive dimensjonene fra Blackwell & Green (2003) og Sharp et al. (2009) presenteres her:

Viskositet: Hvor stor motstand det er mot forandringer. Hvor mye innsats kreves for å utføre én enkelt endring? Hvor mye innsats kreves for å utføre flere endringer av samme type? Vil én endring medføre en dominoeffekt med endringer på andre elementer? Det skilles mellom repetisjonsviskositet, som er flere handlinger av samme type, og konsekvensviskositet, der flere handlinger kreves som et resultat av en handling.

Synlighet: Evnen til enkelt å se komponentene i notasjonen. Er alle delene av notasjonen synlige samtidig? Eller er det mulig å sidestille elementer ved behov? Hvis notasjonen er spredt, er det i det minste tydelig hvilken rekkefølge man skal lese den?

For tidlige forpliktelser: Handlinger og avgjørelser må tas for tidlig i prosessen. Må en ta avgjørelser før man har informasjonen man trenger? Eksempler kan være å måtte velge bestikk før man har bestemt hvilken mat man skal spise, eller å måtte velge hvor mange sider et dokument skal være før man begynner å skrive.

Skjulte avhengigheter: Viktige koblinger mellom entiteter er ikke synlige. Hvilke type avhengigheter er skjulte? Er alle avhengigheter åpenbare i begge retninger? Er avhengighetene perseptuelle eller bare symbolske?

Uttrykksfulle roller: Formålet til en enhet kan enkelt forstås. Er det klart hvordan hvert element er relatert til helheten, og hvilke relasjoner som finnes mellom elementene? Notasjoner som har uttrykksfulle roller gjør det enklere å forstå hvorfor strukturen er på en bestemt måte.

Utsatt for feil: Notasjonen inviterer til å gjøre feil. Kan notasjonens design forårsake slurvefeil? Vil det være enkelt å gjøre feil eller er noen feil vanlige? Sikkerhetsmekanismer

kan forebygge eller forhindre feil, for eksempel å kontrollere antall siffer i et telefonnummer.

Abstraksjon: Typer og tilgjengelighet av abstraksjonsmekanismer. Abstraksjoner, eller redefineringer, endrer den underliggende notasjonen. Kan elementer innkapsles og i hvilken grad? Notasjonen for å ringe med en telefon består av å skrive hele telefonnummeret. En abstraksjon kan være å lagre nummeret som en hurtigtast.

Sekundærnotasjon: Tilleggsinformasjon kan uttrykkes i en annen form enn den formelle syntaksen. Kan det brukes layout, farger eller andre hint for å uttrykke ekstra betydning, i tillegg til å bruke den offisielle notasjonen? Ofte vil brukere ha behov for å notere informasjon som ikke har blitt forutsett av designeren av notasjonen. I stedet for å forutse hvert eneste brukerkrav, støtter flere systemer en sekundærnotasjon som kan bli brukt på den måten som brukeren ønsker.

Relasjonsnærhet: Nærheten til representasjonen i forhold til domenet. Når en ser på notasjonen, er det enkelt å forstå hensikten til hvert element? Hvor direkte kan entiteter i domenet uttrykkes i notasjonen? Inkluderer notasjonen entiteter som kan relateres til nøkkelkonseptene eller nøkkelelementene i domenet? Hvor nært er notasjonen relatert til det resultatet den beskriver?

Konsistent: Like semantikker er uttrykt i lik syntaktisk form. Når deler av notasjon har blitt lært, hvor mye av resten kan utledes? Er lignende struktur og syntakseegenskaper gjennomgående brukt? Brukere tolker ofte strukturen i informasjonsartefakter fra mønstre i notasjonen. Hvis lignende informasjon er presentert på ulikt vis blir brukbarheten svekket.

Diffushet: Ordrikhet ved notasjonen. Hvor mange symboler eller grafiske entiteter kreves for å uttrykke mening? Noen notasjoner kan være irriterende omstendelige, eller ta opp for mye verdifull plass på skjermbildet. Store ikoner og lange ord reduserer det tilgjengelige arbeidsområdet.

Vanskelige mentale operasjoner: Et høyt behov av kognitive ressurser. Bruker notasjonen mekanismer som nøsting og enveisrelasjoner som krever mental dekodning? For eksempel, finnes det steder hvor brukeren må telle på fingrene eller bruke tilleggsannoteringer for å klare å følge med? En notasjon kan gjøre ting komplekse eller vanskelige å forstå, ved å kreve mer av brukernes oppmerksomhet/arbeidsminne enn nødvendig, eller ved å kreve at brukeren utfører dypt nøstede operasjoner for å nå et mål.

Midlertidighet: I hvilken grad forplikter handlinger eller markeringer. Kan midlertidige eller alternative beslutninger uttrykkes? Systemet har mulighet til å gjøre midlertidige handlinger som å lagre potensielle designalternativ, skisseringer og gjøre «hva om» handlinger.

Progressiv evaluering: Oppdatert arbeid kan bli undersøkt på ethvert tidspunkt. Kan en delvis ferdig representasjon testes eller evalueres for å få en tilbakemelding på «hvordan man ligger an»? Systemer kan muliggjøre progressiv evaluering ved å tillate brukere å stoppe halvveis for å vurdere arbeidet som er utført, identifisere fremgang, eller sjekke hvilket stadie av arbeidet de er på.

Da det ikke er mulig at alle dimensjonene oppfylles i like stor grad, må en analyse av kognitive dimensjoner ta hensyn til bruken av notasjonen (Blackwell, 2000b). Grunnen til dette er at ulik bruk krever at ulike dimensjoner er prioritert, noe som medfører at kompromiss mellom dimensjonene, designmanøvre, bør velges på bakgrunn av bruken. Blackwell (2000a) definerer fem kategorier for bruk:

Søking: Finne informasjon ved å navigere gjennom den notasjonelle strukturen (f.eks. finne en spesifikk verdi i et regneark).

Legge til: Legge til ytterligere informasjon i en notasjon, uten å endre strukturen (f.eks. legge til en ny formel i et regneark).

Modifikasjon: Endre en eksisterende notasjonsstruktur, gjerne uten å legge til innhold (f.eks. endre et regneark slik at dette kan brukes til noe annet).

Avskriving: Kopiere innhold fra en notasjon til en annen (f.eks. lese en formel i en bok og konvertere den til en regnearksformel).

Utforskende design: Kombinasjon av kategoriene *legge til* og *modifikasjon*, og samtidig ikke vite på forhånd hva som er målet (f.eks. fortløpende utforme et regneark).

I følge Blackwell (2000b) kan dimensjonene vurderes uten at en streng evalueringprosedyre følges. Studiene av rammeverket presenterer ikke en detaljert prosedyre for hvordan evalueringen skal utføres, men Blackwell (2000b) påpeker at en mulig fremgangsmåte er presentert i Green & Blackwell (1998). Denne fremgangsmåten har fire steg:

1. Identifiser hovednotasjonen ved systemet: beskriv mediet hvor notasjonen er uttrykt, og miljøet hvor notasjonen manipuleres.
2. Identifiser subenheter: hvis det eksisterer subenheter skal disse analyseres som separate notasjoner.
3. Vurder hver notasjon i forhold til dimensjonene.
4. Ved identifisering av brukbarhetsproblemer bør designmanøvrere vurderes for å balansere dimensjonen.

3.4 Oppsummering

I dette kapitlet ble designforskning og dens syv retningslinjer presentert. Videre ble evalueringformene brukbarhetstesting og brukbarhetsinspeksjon beskrevet. Seksjonene beskrev intervju, think-aloud, heuristisk evaluering og det kognitive dimensjonsrammeverket.

4 Forskningsdesign

Relevante forskningsmetoder ble presentert i kapittel 3. I dette kapittelet presenteres studiens bruk av metoder og strategi for å besvare forskningsspørsmålene, presentert i kapittel 1. Dette kapittelet presenterer hvordan studien følger retningslinjene til designforskning. Videre beskrives fremgangsmåten for å samle inn krav og deltakere, utviklingen, analysen og evalueringen til de ulike iterasjonene. Avslutningsvis gjøres det rede for hvordan studien bruker det kognitive dimensjonsrammeverket for å evaluere prototypen.

4.1 Designforskning

Målet med studien er å besvare forskningsspørsmålene presentert i kapittel 1. For å besvare disse er designforskning valgt som overordnet metode. Designforskning er valgt fordi metoden fokuserer på å skape innovative artefakter i den hensikt å svare på spørsmål som er relevante i forhold til menneskelige problem, og dermed bidra med ny kunnskap. Metoden er relevant i denne studien fordi utviklingen og evalueringen av artefaktet gir økt innsikt i hvordan an kan kombinere egenskaper fra fysiske artefakt med programvare. Rammeverket beskriver syv retningslinjer, presentert i seksjon 3.1. Denne studien følger retningslinjene på følgende måte:

1. Design som et artefakt: Artefaktet som utvikles i denne studien er Verdandi. Utviklingen av artefaktet starter med en iterasjon hvor ulike interaksjonsmetoder vurderes. Utvikling og implementasjon av artefaktet beskrives gjennom iterasjoner, som presentert i kapittel 7 og 8.
2. Problemrelevanse: Formålet med studien er å utvikle og evaluere en digital informasjonsradiator for å oppnå økt innsikt i hvordan man kan kombinere egenskaper ved fysiske artefakt og programvare. Som presentert i kapittel 1 kan digitale system tilrettelegge for rapportering, deling av informasjon mellom distribuerte team og integreres med andre systemer.
3. Designevaluering: Artefaktets bruk, kvalitet og virkning vil bli grundig evaluert i form av brukertester i tre iterasjoner, samt evaluert ved en brukbarhetsinspeksjon.
4. Forskningsbidrag: Ved å besvare forskningsspørsmålene og presentere resultatene vil studien være et vitenskapelige bidrag til kunnskapsdomenet.

5. Nøyaktig forskning: Studien følger retningslinjen ved å bruke kjente metoder for utvikling og evaluering av artefaktet, og dette kapittelet er dedikert til å gi innsikt i forskningsprosessen.
6. Design som en søkeprosess: Utviklingen av Verdandi utføres i iterasjoner hvor artefaktet kontinuerlig utvikles og forbedres. Søkeprosessen starter med en konfigurering av MIMIR og vurdering av interaksjonsteknologier og utviklingsrammeverk. Neste iterasjon har som hensikt å brukerteste og evaluere interaksjonsmetodene, og deretter følger to iterasjoner hvor den digitale informasjonsradiatoren utvikles og evalueres.
7. Kommunikasjon av forskningen: Denne studien retter seg i hovedsak mot et teknisk orientert publikum innenfor informasjonsvitenskap, men kan ha relevans fra et forretningsperspektiv som ønsker å ta i bruk en digital informasjonsradiator.

Designforskning er av natur en iterativ metode. Den sjette retningslinjen til Hevner et al. (2004) beskriver design som en søkeprosess hvor man bruker tilgjengelige midler for å oppnå ønsket mål, samtidig som man tar hensyn til begrensningene i problemområdene. Gjennom oppsettet av MIMIR, beskrevet i kapittel 5, og tre iterasjoner med utvikling, er målet å utvikle en digital informasjonsradiator. Informasjonsradiatoren evalueres ved en brukbarhetsinspeksjon og resultatet diskuteres opp mot funnene til Sharp et al. (2009) i kapittel 9.

Søkeprosessen er begrenset til tre iterasjoner med brukertesting på grunn av tidsaspektet ved utviklingen. Underveis i designsyklusen er det beskrevet av Hevner et al. (2004) at man utvikler og evaluerer i iterasjoner, designsyklusen representeres i denne studien som iterasjoner hvor brukertester av systemet står sentralt.

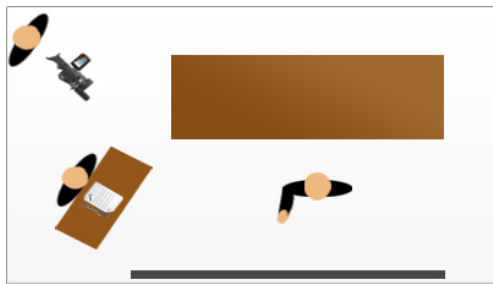
4.2 Brukertesting

Studien benytter brukertesting i iterasjon 1, 2 og 3 for å avgjøre om grensesnittet og interaksjonsmetoden forstås av potensielle sluttbrukere av produktet. Brukertester har som mål å identifisere konkrete brukbarhetsproblemer og innspill som brukes for å forbedre prototypen og få økt innsikt i problemområdet.

Samtlige brukertester vil bli gjennomført i et kontrollert miljø av tre årsaker. For det første tar ikke studien for seg det sosiale perspektivet. For det andre utelukkes variabler fra et naturlig miljø som kan forurensse eller komplisere studien. Den tredje og mer åpenbare

årsaken, er at prototypen utvikles på en videovegg som ikke lar seg å transportere til et naturlig miljø.

Miljøet er et møterom på Universitet i Bergen med tre store vindu og et stort bord i midten av rommet. Videoveggen er montert på den ene langsiden, omtrent en meter fra gulvet. Deltakeren plasseres foran videoveggen, med et par meters avstand til denne og kan fritt bevege seg rundt i rommet. Videokamera plasseres til høyre for videoveggen, men bak deltakeren slik at det er utenfor synsfeltet under brukertesten. Kameraet plasseres på slik for å ha minst mulig påvirkningskraft på deltakeren. Kameraet rettes mot videoveggen og ikke mot deltakeren, både for å få med handlingene som skjer under testen, men også for å formidle at det er prototypen som skal testes, ikke deltakeren. Oppsettet og utsnittet til kamera er vist i figur 6(a) og 6(b).



(a) Figur over laboriemiljøet sett ovenfra med deltaker, primær og sekundærobservatør.

(b) Bilde av utsnittet fra videokameraet.

Figur 6: viser hvordan laboriemiljøet i denne studien var satt opp (a) og (b)

Det er to observatører i rommet. Primærobservatør er ansvarlig for gjennomføringen av testen. Når primærobservatøren ikke demonstrerer funksjonalitet sitter han bak et skrivebord med intervjuguide og skrivesaker, på deltakerens høyre side. Det er primærobservatøren som stiller spørsmålene fra intervjuguiden, og således leder intervjuet og brukertesten. Sekundærobservatør sitter bak kamera, og har i hovedsak som oppgave å kontrollere at video og lyd har tilfredsstillende kvalitet. Sekundærobservatøren kan også hjelpe dersom teknisk utstyr eller prototypen feiler, eller om han finner det nødvendig å stille oppfølgingsspørsmål. Laboriemiljøet brukes i iterasjon 1, 2 og 3.

4.3 Iterasjon 0 - Systemoppsett og valg av interaksjonsteknologier

En iterasjon 0 beskrives av Schuh (2005) som en forberedende fase som ikke direkte gir funksjonalitet, men brukes til å sette opp utviklingsmiljø, rammeverk og systemer som fasiliteter for utviklingen. I denne studiens iterasjon 0 presenteres blant annet oppsettet av MIMIR, hvilken programvare, utviklingsrammeverk og hvilke interaksjonsteknologier som skal benyttes i iterasjon 1, 2 og 3. Som en del av iterasjon 0 vil ulike interaksjonsmetoder innenfor de ulike teknologiene introduseres, og metodene vil evalueres videre i iterasjon 1.

4.4 Iterasjon 1 - Evaluering av interaksjonsmetoder

I den første iterasjonen vurderes ulike interaksjonsmetoder i kombinasjon med MIMIR for å besvare FS2: «Hvilken interaksjonsmetode er best egnet i kombinasjon med MIMIR for å bevare aktiviteten ved en informasjonsradiator?». Målet med iterasjonen er å identifisere en interaksjonsmetode som skal brukes sammen med prototypen. For å kunne gjøre en kvalifisert vurdering ses det som nødvendig å samle inn data fra brukere gjennom en brukertest.

Brukertesten har som mål å belyse interaksjonsmetodenes presisjon, enkelhet og effektivitet, egenskaper som ytterligere beskrives som krav i seksjon 6.1. Brukertesten skal kun belyse selve interaksjonsmetodene, og er ikke avhengig av den digitale informasjonsradiatoren som utvikles. Resultatet kan derfor analyseres uavhengig. Det stilles ingen krav til kunnskap om smidige metoder eller informasjonsradiatorer i denne brukertesten, deltakerne kan derfor velges uavhengig av forkunnskap. Det nødvendige antallet deltakere i brukertesten ble vurdert i forhold til tiden det tok å gjennomføre hver test, samt tiden det tar å analysere materialet. På bakgrunn av dette vurderes seks til åtte deltakere som et kompromiss mellom praktiske forhold og samtidig være et tilstrekkelig grunnlag for å finne en egnet interaksjonsmetode.

Brukertesten består av flere små tester som skal gjennomføres. Utførelsen vurderes etter fire variabler. Det første som registreres er en poengsum som angir hvor godt eller dårlig testen ble gjennomført. Poengsummen registreres av systemet og kan eksempelvis være hvor mange ganger markøren beveger seg utenfor ønsket område. Den andre variabelen som registreres er tid som angir hvor fort deltakeren klarer å gjennomføre testen. En tredje variabel er deltakernes subjektive vurdering av kontrollfølelsen. Vurderingen blir gitt på

en skala fra én til fem, der én er ingen kontroll og fem er full kontroll. Den siste variabelen er uttalelser eller observasjoner som kan underbygge eller forklare resultatene av testen. Dette kan være sitater som «metoden var svært presis, jeg hadde full kontroll», eller en observasjon av at brukeren misforstår oppgaven.

Deltakernes resultater blir ført i et regneark. Samtlige regneark blir deretter sammenslått for å finne gjennomsnittlig poeng, tid, og vurdering. Hvis en test ikke blir gjennomført, eller av en eller annen grunn vurderes som ugyldig, blir den ekskludert. Dersom en deltaker ikke klarer å gjennomføre en test vil kontrollfølelsen settes til én. Relevante sitater eller observasjoner brukes for å begrunne eller underbygge tallmaterialet. Resultatene fra brukertestene presenteres for å kunne vurdere interaksjonsmetodene i forhold til hverandre i seksjon 6.4, og her vil resultatene bli analysert for å danne indikasjoner på styrkene og svakhetene til interaksjonsmetodene. Indikasjonene vurderes for å finne én egnet interaksjonsmetode.

4.5 Iterasjon 2 og 3 - Utvikling og testing av Verdandi

I iterasjon 2 og 3 utvikles og evalueres prototypen. Formålet ved iterasjon 2 og 3 er å utvikle en digital informasjonsradiator som kan sammenlignes med fysiske artefakt. Iterasjon 2 og 3 er metodologisk like, og beskrives derfor i samme seksjon i dette kapittelet. Eneste forskjell mellom de to iterasjonene er at kravene til iterasjon 2 utvikles på bakgrunn av teori, mens kravene til iterasjon 3 stammer fra resultatet av iterasjon 2.

I forhold til prototypen som utvikles kreves det at deltakerne har kunnskap om smidige metoder og artefakter som informasjonsradiatorer, brukerhistorier og burndowndiagram. Ideelle deltakere er derfor personer som har kunnskap og erfaring fra prosjekter, og velges på bakgrunn av denne kunnskapen. Som presentert i seksjon 3.2.3, regnes fire til fem deltakere som tilstrekkelig for å identifisere brukbarhetsproblemer. Antallet settes som et minimumsgrense, med mulighet for å utvidelse dersom det er tid og ressurser til rådighet.

For å evaluere artefaktene som utvikles, er brukertesting med think-aloud valgt som evalueringsmetode. Evalueringsmetoden gir et innblikk i hva deltakerne tenker og hvordan de tolker ulike elementer ved artefaktet. Mer spesifikt er think-aloudrammeverket til Boren & Ramey (2000) valgt for å rettlede sesjonene i form av definerte rammer, språkbruk, oppførsel samt det å få fram ytterligere informasjon. Listen nedenfor beskriver

hvordan retningslinjene følges i denne studien:

Definerte rammer: Under evalueringen av iterasjonene er det tatt hensyn til denne retningslinjen ved at alle deltakerne får opplest en introduksjonstekst som setter rammene for brukertesten. Som forklart i seksjon 4.2 er plasseringen av teknisk utstyr i henhold til denne retningslinjen for ikke å være forstyrrende for deltakeren.

Språkbruk: For å oppfordre deltakeren til å kontinuerlig verbalisere sine tanker er det gitt korte anerkjennelser, slik som beskrevet i Boren & Ramey (2000). Anerkjennelsene gis i form av korte bekreftelser som «Mm hm».

Oppførsel: Kommunikasjon mellom observatør og deltaker brukes når dette er nødvendig. En mal for å rettlede brukertesten blir brukt i iterasjonene for å unngå emner som er utenfor interesseområde, samt for å belyse aktuelle emner. Under brukertestene må teknisk utstyr kalibreres, noe som i følge Boren & Ramey (2000) er akseptabelt dersom det gjøres klart at inngrepet ikke er en del av brukertesten.

Få fram ytterligere informasjon: I forhold til denne retningslinjen benyttes det en mal for å rettlede brukertesten i iterasjonene. Oppfølgings spørsmål blir stilt når det er nødvendig, for å forsikre at informasjonen fra deltakeren er korrekt oppfattet.

I tillegg til think-aloud er det i hver brukertest brukt semistrukturert intervju for å kontrollere deltakernes kjennskap til informasjonsradiatorer. Denne kombinasjonen av metoder kan gjøre det mulig å se deltakernes uttalelser i lys av deres kunnskap, samt sammenligne med andre deltakere. Intervjuet er kort og konsist, og vil kun ha som mål å kontrollere at deltakeren har kjennskap til hovedelementene ved informasjonsradiatoren. Semistrukturert intervju brukes for å ha muligheten til å stille oppfølgings spørsmål der noe er uklart. I tillegg ble bilder eller tegninger av deltakernes informasjonsradiator samlet inn for å supplere intervjuene.

Brukertestene ble transkribert, og deretter lagt til metainformasjon som inndeling av spørsmål, hvem som snakker, hva som er handlinger og kommentarer. For å legge til metainformasjonen blir teksten formatert i et XML-dokument¹⁵, med tilhørende tagger og attributter. Ved å gjennomføre denne transformasjonen blir dokumentet søkbart med

¹⁵XML: eXtensible Markup Language er et markeringsspråk med mulighet for å definere egne tagger.

XPath¹⁶ og XQuery¹⁷, og det blir enklere å legge til ytterligere metainformasjon ved en senere anledning.

For å markere uttalelser som er mulige brukbarhetsproblemer i dokumentet blir det i hver iterasjon opprettet et kategoriskjema. Skjemaet opprettes ved at en delmengde av datamaterialet blir gjennomgått individuelt og mulige kategorier noteres. Identifiserte kategorier sammenlignes, diskuteres og samskrives til ett kategoriskjema. Dette skjemaet kontrolleres for overlappende kategorier, noe som i følge Sharp et al. (2011) øker påliteligheten ved kategoriene, som igjen gjør det mulig for andre å gjenta kategoriseringen. Kategoriseringen ble i iterasjon 2 og 3 utført individuelt for å øke sannsynligheten for å identifisere uttalelser som er mulige brukbarhetsproblemer. Etter den individuelle kategoriseringen ble resultatene sammenlignet for å eliminere uvesentlige ulikheter, som f.eks. feilmarkering av teksten eller hvis begynnelsen og slutten av en uttalelse er markert ulikt.

Uttalelser som er mulige brukbarhetsproblemer blir markert i forhold til kategoriskjemaet ved å legge til tagger på formen: <u type=X.Y> der X.Y er nummeret på brukbarhetskategorien i kategoriskjemaet. Figur 7 viser et eksempel på hvordan kategoriseringen ble utført.

<pre>- Du snakket om disse pilene opp og ned i begynnelsen. Testpersonen prøver å trykke på knappene - Den var ikke noe god her borte. Pilene var helt forferdelig. Det var veldig løye - Løye som i gøy, eller løye som i rart? - Løye som i rart. Latter</pre>	<pre><s4.2> <spm>Du snakket om disse pilene opp og ned i begynnelsen.</spm> <u type="5.2"> <handling>Testpersonen prøver å trykke på knappene</handling> Den var ikke noe god her borte. Pilene var helt forferdelig. Det var veldig løye </u> <spm>Løye som i gøy, eller løye som i rart?</spm> Løye som i rart. <notat>Latter</notat> </s4.2></pre>
--	---

Figur 7: viser transkriptet før og etter at metainformasjon er lagt til. Forklaring av annoteringer: <s4.2> beskriver hvilket spørsmål innholdet omhandler; <spm> er spørsmålet slik det ble stilt av observatør; <u type="5.2"> beskriver et mulig brukbarhetsproblem av typen 5.2 fra kategoriskjemaet.

¹⁶XPath - Et spørrespråk for XML-dokument

¹⁷XQuery - Et spørrespråk for XML-dokument utvidet med programmeringsfunksjonalitet. XPath er en del av XQuery.

For å skape en bedre oversikt samt for å forenkle analysen av datamaterialet, ble identifiserte brukbarhetsproblemer slått sammen og sortert etter kategori og deltaker. Denne operasjonen ble utført med XQuery-spørringen som er vist i figur 8:

Resultatet av sammenslåingen er en oversikt av alle uttalelser som er mulige brukbarhetsproblemer. Ved å se etter mønstre, tema og kategorier kan oversikten brukes for å evaluere prototypen. Resultatet av evalueringen er, i iterasjon 2, krav til videre utvikling, mens i iterasjon 3 er resultatet funn som vurderes opp mot den kognitive dimensjonsanalysen.

```
<root>{
for $u in distinct-values (//u/@type)
  order by $u
  return (<brukbarhetsproblem nr="{ $u }">{for $p in //person
  where count ($p/u[./@type = $u]) > 0
  return (<person>{ $p/nr }<antall>{ count ($p/u[./@type = $u]) }
  </antall>{ $p/u[./@type = $u] }</person>
  }</brukbarhetsproblem>
}</root>
```

Figur 8: viser et XQuery-uttrykk for å slå sammen og sortere brukbarhetsproblem etter kategori og deltaker.

4.6 Kognitiv dimensjonsanalyse

Som presentert i seksjon 2.1.5 benytter Sharp et al. (2009) et rammeverk for å beskrive fysiske brukerhistoriekort og informasjonsradiatorer i forhold til kognitive dimensjoner. Ved å benytte samme metode kan det dannes et grunnlag for sammenligning av dimensjoner mellom fysiske og digitale versjoner av artefaktene. Sammenligningen har som mål å besvare FS1: «Hvordan endres notasjonsperspektivet ved informasjonsradiatoren når den digitaliseres på MIMIR?». Siden Sharp et al. (2009) har analysert fysiske informasjonsradiatorer ved hjelp av kognitive dimensjoner, er det naturlig å utføre en tilsvarende analyse av Verdandi. Mulighet for å sammenligne resultatene er årsaken til at det kognitive dimensjonsrammeverket foretrekkes fremfor andre metoder, som heuristisk evaluering. Analysen av Verdandi følger fremgangsmåten som presenteres i seksjon 3.3.2 på følgende måte:

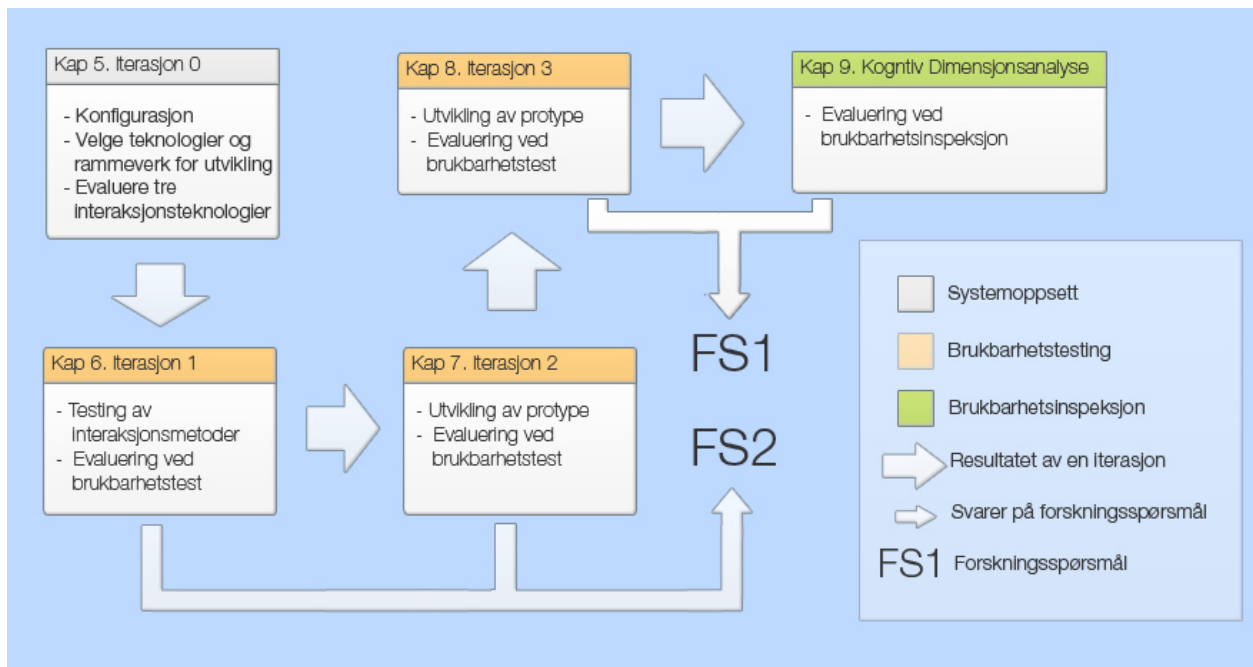
1. Hovednotasjonen, medie og miljø: Verdandi som helhet er hovednotasjonen som analyseres og beskrives i seksjon 8.1. Mediet er MIMIR og beskrives i seksjon 5.1.

- Miljøet hvor Verdandi manipuleres er som beskrevet i seksjon 4.2.
2. Identifiser subenheter: Steget er ikke relevant, Verdandi har ingen subenheter.
 3. Notasjonene i forhold til dimensjonene: Dette steget er hvor hoveddelen av rammeverket realiseres. En vurdering av alle dimensjonene er presentert som en del av kapittel 9. Dimensjonene blir også diskutert i forhold til hverandre i samme kapittel.
 4. Designmanøvre: Designmanøvre som kan balansere forholdet mellom de ulike dimensjonene beskrives i seksjon 9.2.

Dimensjonsanalysen utføres etter den tredje iterasjonen, og resultatet av analysen vil være en beskrivelse av 14 dimensjoner samt retningslinjer for å utvikle digitale informasjonsradiatorer.

4.7 Oppsummering

I dette kapittelet har forskningsdesignet for denne studien blitt presentert. Det fremgår av kapittelet at designforskning er brukt som overordnet metode, hvor det gjennom designforskningen utvikles et artefakt, Verdandi, i tre iterasjoner. Brukertestene i hver iterasjon utføres i samme laboratoriemiljø, og fokuserer på å kontinuerlig forbedre artefaktet. Kapittelet avsluttes ved å presentere en kognitiv dimensjonsanalyse, en brukbarhetsinspeksjonsmetode, som brukes for å evaluere prototypen. Prosessen illustreres i figur 9.



Figur 9: viser en grafisk fremstilling av forskningsdesignet og hvordan de ulike iterasjonene er relatert.

5 Iterasjon 0

En forutsetning for å kunne utvikle Verdandi er at MIMIR settes opp, samt at kandidater til interaksjon identifiseres for videre brukertesting. I tillegg må det tas et valg av utviklingsrammeverk og teknologier for å utvikle prototypen. Fordi MIMIR er et komplekst system blir det gitt en beskrivelse av hvordan det ble satt opp og konfigurert. Det vil bli presentert valg av operativsystem, samt relevante drivere og programvare. Videre vil det følge en vurdering av ulike interaksjonsteknologier som kan brukes med Verdandi, og avslutningsvis presenteres utvalgte utviklingsrammeverk og tilhørende teknologier.

5.1 Systemoppsett og konfigurasjon av MIMIR

MIMIR er en skjermbasert videovegg hvor man, via 18 HD-skjermer, har muligheten til å vise store mengder informasjon. Hver av de 18 skjermene til MIMIR er 24 tommer, og totalt er MIMIR 95 cm høy og 330 cm bred, oppsettet vises i figur 2. Til et system av denne størrelsen kreves det ulike tilpasninger av maskinvare og programvare. Maskinvarespesifikke tilpasninger er realisert gjennom en sekskjerners prosessor, tre ATI Radeon HD 5870 Eyefinity skjermkort¹⁸, 12 gb minne og to SSD¹⁹-harddisker. På programvarenivå er de største forskjellene merkbare i ulike operativsystem med ulike drivere, noe som beskrives nærmere i seksjon 5.1.1.

Bruksområdet til MIMIR var ment å støtte visualisering av store datasett, muliggjøre analysering av store mengder bildedata, samt by på interessante problemstillinger innen interaksjonsdesignforskning (Gjøsæter & Krzywinski, 2012). MIMIR var imidlertid ikke ferdigbygd da denne studien startet, og bestod av bare av ni skjermer. Som det vil fremgå av neste seksjon, er det først når man går over ti skjermer at konfigurasjon og oppsett av systemet blir problematisk. Etter at systemet ble bygd ferdig, ble en utforskende vurdering av Microsoft Windows 7 og Linux distribusjonen Ubuntu Skrivebord 11.04 utført. Vurderingen presenteres i neste seksjon.

¹⁸ATI Radeon HD 5870 Eyefinity er et skjermkort med utganger til seks skjermer. Kortet var det første som klarte å rendre 1 mrd piksler per sekund og regnes som svært ytelsessterkt (Advanced Micro Devices, 2010)

¹⁹SSD, Solid-State Drive er en harddisk som benytter flashminne fremfor mekanisk bevegelige magnetplater. SSD er derfor vesentlig raskere enn tradisjonelle harddisker

5.1.1 Windows versus Linux

Windows 7 sin standard monitorkonfigurasjon viste seg å ikke håndtere mer enn ti skjermer, noe som skapte problemer når skjermene skulle konfigureres i forhold til hverandre. Denne begrensningen viste seg tydelig i operativsystemets meny for skjerminnstillinger hvor menyen ikke klarte å identifisere skjerm 11 til og med skjerm 18.

På grunn av manglende konfigurasjonsmuligheter i Windows 7 ble det valgt å benytte et tredjepartsprogram. Tredjepartsprogrammet Ultramon støtter videovegger og gjorde det mulige å konfigurere skjermenes posisjon. Imidlertid viste det seg at systemet var ustabil når Ultramon var aktivt. Skjermer slo seg av etter en vilkårlig tid og oppløsningen på noen skjermer var lavere enn definert. En feilsøkningsprosess startet som følge av dette. I første del av denne feilsøkningsprosessen ble ingen tilfredsstillende løsning funnet. Windows 7 ble derfor satt til side, til fordel for Linux.

Skrivebordsversjonen av Ubuntu 11.04 virket lovende i forhold til konfigurasjon av skjermoppsettet, da man i Linux har muligheten til å skrive konfigurasjonsfilen for skjermoppsett manuelt og fordi det ikke ble identifisert en programvarebegrensning på antall skjermer. Konfigurasjonsfilen som ble skrevet og lastet inn, ble vurdert som stabil, men ikke tilfredsstillende. Grunnen til dette var X11²⁰ sin klient-tjener implementasjon hvor hver skjerm er tilknyttet en egen klient, noe som gjorde at denne løsningen krevde for mye minne. Minnebruken førte til at systemets ytelse ble vesentlig dårligere enn i Windows 7. Videre ble det identifisert at proprietære drivere utgitt av ATI var overlegne de åpne kildekodedriverne når det gjelder rendring²¹ av elementer. At de proprietære driverne var overlegne var tydelig ved flytting av elementer mellom skjermer koblet til ulike skjermkort. Flyttingen av elementer opplevdes som «hakkete» og forsinket. Den lave systemytelsen i kombinasjon med liten utviklingsaktivitet rundt Microsoft Kinect og Nintendo Wii på Linux førte til at Windows 7 ble testet på nytt.

I Windows 7 ble det, etter en omstendelig feilsøkningsprosess, funnet at ATI sin konfigurasjonsapplikasjon, Catalyst Control Center, skapte problemer for Ultramon. Catalyst Control Center ble derfor fjernet fra systemet, noe som resulterte i at ustabiliteten ved skjermkonfigurasjonen forsvant.

²⁰X11: X Window System, vanligvis kalt X, er høyt konfigurerbart, kryssplattform, komplett og lisensfri klient-serversystem for å håndtere grafiske brukergrensesnitt på en eller et nettverk med datamaskiner. Tallet i X11 står for versjonsnummeret (The Linux Information Project, 2006).

²¹Eng.: Rendering

Som en følge av at ATI sitt kontrollcenter ble fjernet uteble muligheten til å justere skjermkortenes viftehastighet og skjermkortenes standardinnstillinger ble brukt, noe som betyr at kun en vifte om gangen var aktiv og systemet ble overopphetet. For å aktivere skjermkortviftene ble en rekke vifte- og overklokkingsprogrammer testet, blant annet Rivatuner²², Fancontrol²³, Speedfan²⁴ og MSI Afterburner 2.1.0²⁵. Ingen av disse programmene klarte å styre alle tre skjermkortviftene, dette til tross for at støtten var annonsert på deres respektive nettsider. Løsningen på vifteproblematikken kom da MSI Afterburner Beta 2.11 v6 ble testet. MSI Afterburner var det eneste identifiserte programmet som hadde støtte for tre skjermkort av typen som står i MIMIR. Microsoft Windows 7 ble dermed valgt som operativsystem. MIMIR på programvarenivå var dermed satt opp med:

- Microsoft Windows 7 64 bit - Operativsystem.
- ATI Eyefinity drivere uten Catalyst Control Center 64 bit - Drivere til skjermkortene.
- Ultramon - Konfigurasjonsverktøy for skjermene.
- MSI Afterburner Beta 2.11 v6 - Viftekontroll.

5.2 Interaksjonsmetoder

Som beskrevet i seksjon 2.1.1 trekker Sharp et al. (2009) frem en rekke studier som viser at det fysiske mediet påvirker koordinering og samarbeid ved flere aktiviteter, og at en nøkkelsekvens ved den fysiske naturen til artefaktene er at aktiviteten ikke enkelt kan skjules. Det stilles derfor krav om at interaksjonsmetodene i Verdandi skal være synlige, slik at aktiviteten foran informasjonsradiatoren ikke skjules. På grunn av at Ball & North (2005) betegner tradisjonell interaksjon med mus som uegnet på videovegger og fordi denne metoden er mindre synlig i forhold til andre metoder, forkastes metoden. I tillegg til synligheten tas det hensyn til tilgjengeligheten og kostnadsrammene ved interaksjonsteknologiene. På bakgrunn av disse forholdene vurderes tre teknologier som potensielle kandidater: gyromus, Microsoft Kinect og Nintendo Wii.

²²Rivatuner: <http://www.guru3d.com/category/rivatuner/>

²³Fancontrol: <http://www.lm-sensors.org/wiki/man/fancontrol>

²⁴Speedfan: <http://www.almico.com/sfdownload.php>

²⁵MSI Afterburner 2.1.0: <http://event.msi.com/vga/afterburner/index.htm>

5.2.1 Gyromus

En gyromus er en datamus som benytter seg av et gyroskop for å bevege musepekeren. I systemet benyttes Gyration Air Mouse²⁶. Gyromusen er i utgangspunktet utviklet for presentasjoner på projektorsystemer, men vurderes etter initielle tester til å fungere tilfredsstillende på MIMIR. Gyromusen vurderes ytterligere som svært presis og samtidig enkel å bruke, og er derfor beholdt som en mulig interaksjonsmetode.

5.2.2 Microsoft Kinect

Microsoft Kinect er en kameramodul som består av et RGB-kamera og en IR²⁷-modul. IR-modulen fungerer som et avstandskamera ved at den kontinuerlig projeksjoner strukturert infrarødt lys (Gallo et al., 2011). Ved hjelp av dette lyset kan man tolke tredimensjonal informasjon som for eksempel en persons posisjon.

Det finnes flere teknikker for å tolke informasjonsstrømmen fra Kinect. Blant annet kan man tolke informasjonen ved hjelp av Microsoft sitt programmeringsgrensesnitt for Kinect. Løsningen fra Microsoft ble nøye vurdert, men valget falt på FFAST²⁸ fordi FFAST har et skriptspråk som er enklere og mer effektivt å benytte enn programmeringsgrensesnittet til Kinect. FFAST er en programvare som benytter seg av skjelettgjenkjenning ved hjelp av OpenNI²⁹ og NITE³⁰ for å tolke datastrømmen som tilgjengeliggjøres fra Kinect (Suma, Lange, Rizzo, Krum & Bolas, 2011). OpenNi-modulen gjør det mulig å kartlegge hvor ulike kroppsdelene er i forhold til hverandre til en hver tid. For at FFAST skal kunne gjenkjenne de ulike kroppsdelene må programvaren kalibreres før bruk. Kalibreringen gjøres ved at man står foran kameraet med skuldre og armer i 90 graders vinkel som vist i figur 10, helt til en strekfigur vises på skjermen. Når kalibreringen er fullført er det mulig å få kontroll over markøren, samt koble gester til funksjoner via det innebygde skriptspråket. FFAST benyttes for å lage fire egendefinerte interaksjonsmetoder, metodene presenteres her:

²⁶Gyration Air Mouse: en gyromus som er ment å brukes til presentasjoner og i andre situasjoner hvor man står unna datamaskinen

²⁷IR: InfraRød

²⁸FFAST: Flexible Action and Articulated Skeleton Toolkit

²⁹OpenNI - API som støtter utvikling av applikasjoner som benytter naturlig interaksjon.

³⁰NITE - API for 3D tolkning. NITE er mellomvaren mellom Kinect og OpenNI



Figur 10: viser posituren for å kalibrere FAAST-programvaren. Den blå skyggen er det infrarøde bildet av brukeren, den røde strekfiguren er skjelettet som vises etter en vellykket kalibrering.

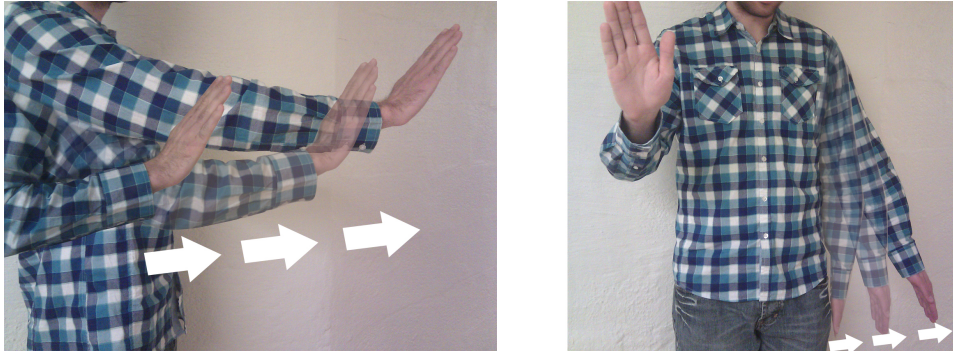
Pushmetoden I denne metoden styrer og velger man elementer med én og samme hånd. Man styrer markøren ved å bevege håndflaten i horisontal og vertikal retning, mens for å velge elementer beveger man håndflaten i dybderetningen. Dette er illustrert i 11(a).

Pingvinklikkmetoden Denne interaksjonsmetoden er utviklet som en del av denne studien, og ble først oppdaget under evalueringen av programvaren FAAST i iterasjon 0. I utgangspunktet var dette bare en metode utviklet for å teste programvaren, men viste seg å være overraskende funksjonell, og er derfor beholdt. I denne metoden styrer man musepekeren med en hånd og velger elementer med den andre hånden. Styringen av musepekeren foregår på samme vis som i pushmetoden, men valg av elementer skjer ved at man tar den andre hånden 20 cm ut fra låret, som illustrert i 11(b).

Interaksjonsmetode med gester Med denne interaksjonsmetoden menes handlinger som å vinke, holde opp armen, gjøre en sirkelbevegelse o.l. FAAST støtter for øyeblikket kun enkle gester som å holde opp eller ut en arm.

Interaksjonsmetode med hoverering Navnet til interaksjonsmetoden kommer fra det engelske begrepet «hover», som betyr og være over noe. Metoden baserer seg på å gjøre handlinger ved å holde markøren over et område i en gitt tid. Et eksempel på metoden kan være å holde over en knapp i et sekund for å trykke på den.

Etter initiell testing og vurdering virker interaksjonsteknologien Microsoft Kinect som stabil og enkel å bruke. Teknologien, med de metodene som er definert, vurderes til å fungere tilfredsstillende, og beholdes for videre testing.



(a) Illustrerer hvordan man velger med pushmetoden. (b) Illustrerer hvordan man velger med pingvinklikkmetoden.

Figur 11: illustrere to interaksjonsmetoder med Kinect, pushmetoden i figur (a) og pingvinklikkmetoden i figur (b)

5.2.3 Nintendo Wii

Spillkonsollen Nintendo Wii styres med en kontrollenhet som inneholder en IR-sensor og et akselerometer. IR-sensoren brukes til å registrere infrarødt lys fra en lyskilde som plasseres foran en tv-skjerm hvor spillkonsollens grensesnitt vanligvis vises.

En eller flere kontrollenheter kan kobles til MIMIR og benyttes til å registrere IR-lys fra IR-penner³¹. Ved å plassere to IR-sensorer som peker mot MIMIR og ved å benytte programvare som Smoothboard³², kan lyset fra IR-pennene tolkes av programvaren og simulere berøringsinteraksjon på MIMIR. Å bruke IR-sensorene sammen med datamaskiner for å simulere berøringsinteraksjon er ikke nytt, blant annet har Lee (2008) gjort dette tidligere.

For å vurdere om teknologien er en potensiell interaksjonsteknologi ble det utført en vurdering av teknologiens dekningsgrad, samt dens evne til å registrere IR-lys fra ulike avstander og vinkler. Med dekningsgrad menes det hvor stort areal som dekkes av en IR-sensor.

Dekningsgrad: For å kunne interagere med hele skjermflaten til MIMIR ble det tidlig i prosessen identifisert at én IR-sensor ikke var tilstrekkelig. Det ble derfor forsøkt å bruke to IR-sensorer. For å kunne kalibrere to IR-sensorer kreves det et overlapp mellom sensorene,

³¹IR-penn: en infrarødt lyskilde som er utformet som en penn.

³²Smoothboard er en programvare som gjør det mulig å simulere berøring med opptil to Nintendo Wii kontrollenheter og IR-penner. Smoothboard er tilgjengelig på <http://www.smoothboard.net/>

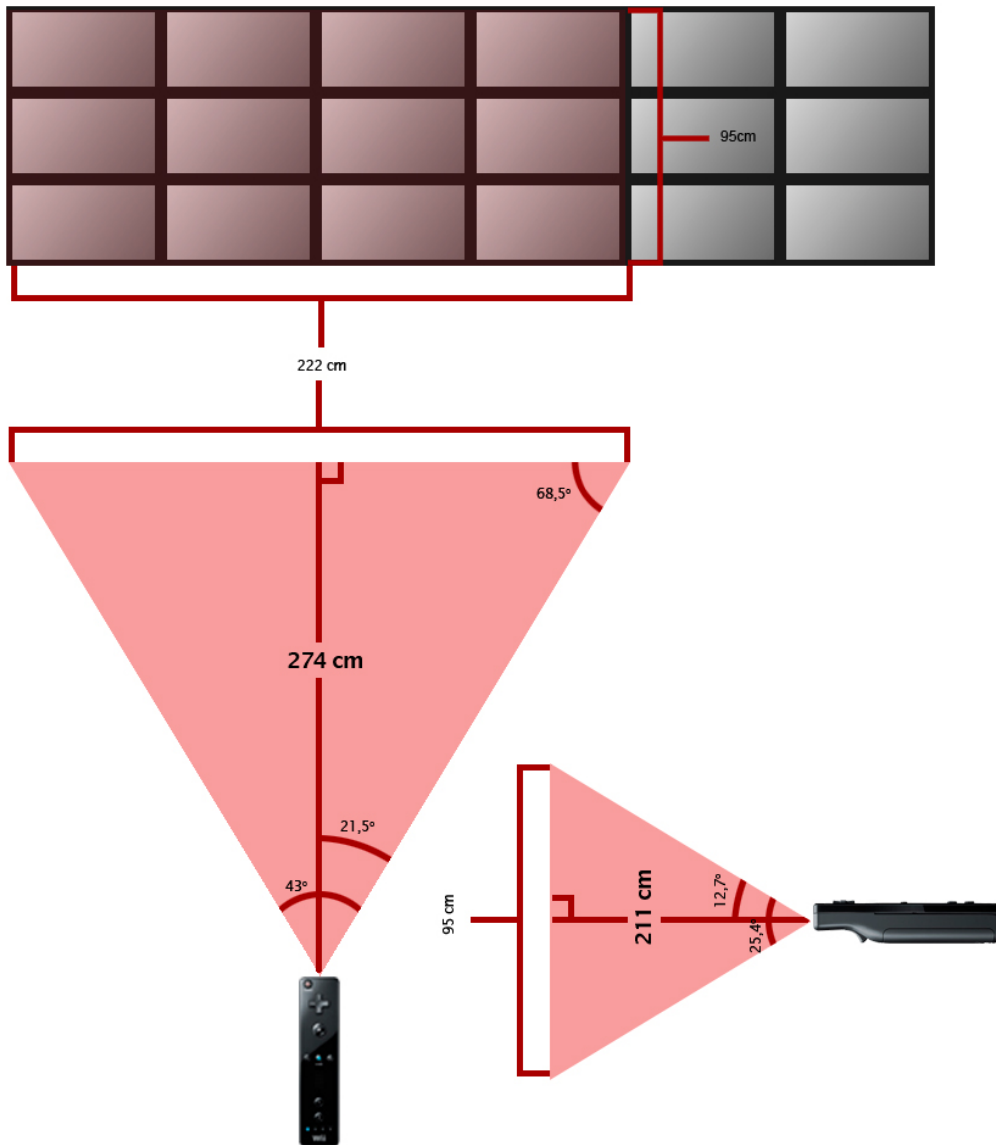
og forsøk viste at overlappet måtte være på minst to skjermkolonner.

Med disse begrensningene viste forsøk at optimal avstand mellom IR-sensor og MIMIR var 2,74 meter for å få tilstrekkelig horisontal dekningsgrad på 4x3 skjermer. Grunn til at avstanden måtte være 2,74 meter er at sensorens horisontale synsfelt er ca. 43 grader. Ved å plassere IR-sensoren 2,74 meter fra MIMIR dekket IR-sensoren 29 cm mer enn nødvendig i vertikal retning, dette gav en total utnyttelsesgrad per IR-sensor på 77%. IR-sensorenes dekningsgrad illustreres i figur 12.

Lysrefleksjon: For at teknologien skal fungere er det nødvendig at IR-sensoren registrerer lyset fra IR-pennene, på grunn av dette ble sensorens evne til å registrere IR-lys på ulike avstander og ved ulike vinklinger testet.

Det ble identifisert at IR-sensoren kan registrere direkte IR-lys på avstander lengre enn fem meter. Når lyskilden peker mot et whiteboard er denne avstanden marginalt kortere. Refleksjonen fra MIMIR, ved vanlig bruk av IR-pennen, var på mindre enn én meter og ikke i nærheten av refleksjonen som skapes av et whiteboard.

Forkastelse: På bakgrunn av resultatene fra dekningsgrad og lysrefleksjon, ble det besluttet å forkaste Nintendo Wii som interaksjonsteknologi da begrensningene ved denne teknologien var for store. Faktumet at lyskilden vinkles bort fra sensoren i MIMIR er hovedproblemet med denne interaksjonsteknologien. Begrensningene relatert til sensorens evner til å registrere IR-lys over større flater var også et problem, da dette vil gjøre metoden unøyaktig. Teknologien vurderes dermed som *uegnet* som interaksjonsteknologi på videovegger som MIMIR.



Figur 12: Ved en avstand mellom sensor og skjerm på 2,74 meter var utnyttelsesgraden 100% langs horisontalaksen og 64,7% langs vertikalaksen. For å få 100% utnyttelse av sensoren ved vertikalaksen skulle sensoren vært plassert 2,11 meter fra skjermen. Med avstand på 2,11 meter, ville dekningsområdet langs horisontalaksen vært for liten til at alle 18 skjermene var dekket av de to sensorene.

5.3 Utviklingsrammeverk

Utvikling av programvare som kjøres på MIMIR kan være komplisert, da man på grunn av de tre ulike skjermkortene programmatisk må synkronisere når ulike elementer skal vises på skjermene. Hvis ikke vil elementer som vises over flere skjermer være usynkronisert, hvilket vil være svært synlig når elementer er i bevegelse.

Siden Verdandi er et forholdsvis statisk system med få bevegelige elementer er det besluttet å utvikle systemet som en spesialtilpasset nettside. Å implementere Verdandi som en nettløsning gjør utviklingen enklere enn om man skulle tatt hensyn til synkronisering av elementer. Utviklingen blir utført ved hjelp av Play!, som er et rammeverk for utvikling av webapplikasjoner i Java og Scala. Rammeverket integrerer blant annet Hibernate³³, jQuery³⁴ og en Groovy templatmotor³⁵ for frontend utvikling. Play! applikasjoner kan brukes på en rekke webservere, gir muligheten for hurtig utvikling og krever minimalt med konfigurasjon. Rammeverket passer dermed bra som utviklingsverktøy for Verdandi.

På grunn av Play! sin innebygde Hibernate-modul var databasedesign ikke en del av utviklingen, da Hibernate ble benyttet til å lagre alle datamodeller. For å ha likt utgangspunkt hver gang Verdandi startes blir det benyttet en cron-jobb³⁶ for å fylle datamodellen ved applikasjonsstart. Siden Play! har full støtte for cron-jobber vil rammeverket ta seg av nødvendige databasetilkoblinger i denne sammenheng. Videre inneholder Play! rammeverket en YAML³⁷ parser, som er en menneskevennlig standard for dataserialisering. Parseren ble benyttet for å fylle den initielle cron-jobben og templatmotoren ble benyttet for å få et modulært kodedesign.

Webapplikasjonen ble kjørt i nettleseren Google Chrome, da dette var den eneste identifiserte nettleseren som kjørte stabilt på alle 18 skjermene samtidig. Av nettlesere som ble forsøkt var Mozilla Firefox, Internet Explorer og Opera. Interaksjonsmetodene i seg selv trengte ingen spesifikk utvikling, da gyromusen kun trengte konfigurasjon og Microsoft Kinect kun trengte enkel konfigurasjon ved hjelp av skriptspråk.

³³Hibernate: Databasehåndteringsverktøy for Java

³⁴jQuery: JavaScript bibliotek som forenkler skripting på klientsiden

³⁵Groovy templatmotor: Templatsystem som gjør det mulig å lage forhåndsdefinerte maler for å dynamisk generere HTML, XML, JSON og tekstbasert innhold

³⁶Cron-jobb: Tidsbasert arbeidsoppgavehåndtering

³⁷YAML: YAML Ain't Markup Language

5.4 Oppsummering

Følgende kapittel har beskrevet hvordan MIMIR er satt opp og konfigurert med Windows 7, ATI Eyefinity drivere, Ultramon og MSI Afterburner. Utviklingsrammeverket Play! med tilhørende teknologier er valgt som plattform for utvikling av prototypen. Tre interaksjonsteknologier ble vurdert, Microsoft Kinect, gyromus og Nintendo Wii. Sistnevnte ble forkastet da teknologien ikke var egnet sammen med MIMIR, de to andre vurderes som egnet for evaluering i form av brukertester.

6 Iterasjon 1

I denne iterasjonen er hovedfokuset å identifisere én interaksjonsmetode som vil bevare aktiviteten foran informasjonsradiatoren. Samtidig stilles det krav til metoden i form av *presisjon, enkelhet og effektivitet*. Interaksjonsmetodene vil bli brukertestet, og resultatene av disse testene vil bli analysert og presentert. Videre vil resultatene danne et designgrunnlag for utforming av Verdandi. Avslutningsvis vil kapittelet oppsummeres.

6.1 Interaksjon

Som det fremgår av iterasjon 0, kapittel 5, er de aktuelle interaksjonsteknologiene begrenset til Microsoft Kinect og gyromus. Disse teknologiene har ulike egenskaper og opprinnelige bruksområder. Som beskrevet i seksjon 5.2.2 kan teknologien Microsoft Kinect brukes på flere måter, og dermed fasilitere for ulike interaksjonsmetoder. For å finne den best egnede interaksjonsmetoden til Verdandi, stilles det følgende krav:

Krav 1: Interaksjonsmetoden bør være *presis*. Presisjonen ved interaksjonsmetoden er dens evne til nøyaktig å gjengi den handlingen som brukeren prøver å utføre.

Krav 2: Interaksjonsmetoden bør være *enkel*. Med enkel menes det her at brukeren skal forstå og ha kontroll over interaksjonsmetoden. Kontroll er oppnådd når brukeren klarer å utføre det han ønsker.

Krav 3: Interaksjonsmetoden bør være *effektiv*. Med effektiv menes her at man hurtig kan bevege markøren over hele skjermflaten samt velge og flytte elementer.

6.2 Interaksjonstester

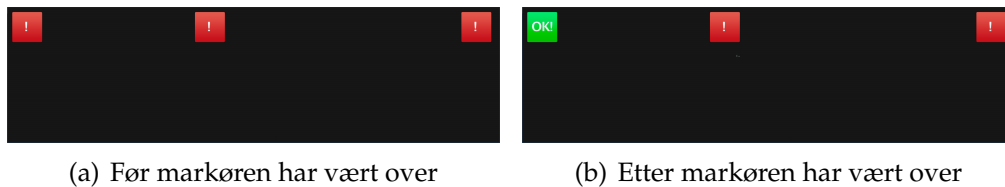
For å identifisere hvilke interaksjonsmetoder som er best egnet, ble det utviklet et sett med interaksjonstester. Disse testene ble utformet slik at man kan danne indikasjoner om interaksjonsmetodene i forhold til kravene som er stilt. I tillegg er det utformet en gesttest som spesifikt er beregnet på Microsoft Kinect.

Totalt ble det utviklet seks interaksjonstester. Testene måler ulike variabler som tidsbruk, presisjon og antall forsøk. I alle interaksjonstestene vil en subjektiv kontrollfølelse vurderes på en skala fra én til fem.

6.2.1 Knappetest

Knappetestens formål er å måle interkasjonsteknologiernes *stabilitet* og *nøyaktighet*, dvs. teknologiens evne til å gi samme resultat gjentatte ganger, samt hvor nøyaktig markørkontroll teknologiene gir.

Testen er utformet ved at det er flere kvadratiske knapper fordelt over skjermene, som illustrert i figur 13(a). Knappene posisjoneres ved ytterpunktene av skjermflaten for å kunne vurdere hvordan interaksjonsmetodene takler å navigere i skjermenes ytterpunkter, samt å tvinge brukeren til å navigere over store områder. To sett med knappetester er utviklet, det ene testsettet er med knapper på øverste skjermrekke, det andre testsettet er med knapper på nederste skjermrekke.



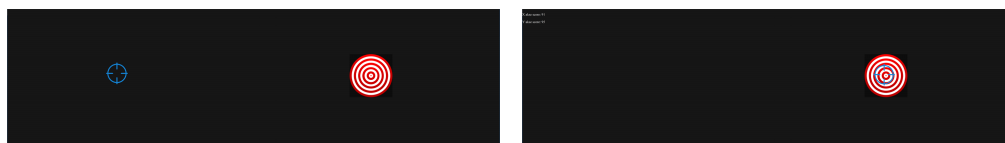
Figur 13: viser knappetest på øverste skjermrekke før markøren har vært over (a) og etter markøren har vært over (b)

Oppgaven til deltakeren er å holde markøren i ro over hver knapp i to sammenhengende sekunder. Når dette er fullført skifter fargen fra rød til grønn, illustrert i figur 13(b). For å få en indikasjon på hvilken størrelse elementer i prototypen kan ha, kjøres knappetesten i tre versjoner, hvor størrelsen på knappene blir gradvis mindre i hver test. En test ble definert som ferdig når deltakeren har klart å holde markøren over samtlige knapper. Deltakeren vil da få en score som forteller hvor mange ganger markøren forlot knappen i løpet av testen, som vil gi en indikasjon på stabiliteten til interaksjonsmetoden.

6.2.2 Blinktest

Blinktestens formål er å måle hvor *enkelt* det er å plukke opp, flytte, og slippe et element på et ønsket område. Deltakerne skal plukke opp et sikte og slippe det så sentrert som mulig over en blink. Testene sitt utgangspunkt vises i figur 14(a) og en fullført test vises i 14(b) med siktet plassert over blinken. Utifra hvor presist deltakeren har plassert siktet vil det bli gitt en poengsum. Testen blir gjennomført i to utgaver, der den siste utgaven har mindre

sikte og blink. Det forventes at nivået til presisjon og stabilitet vil bli mer utslagsgivende i den siste testen.



(a) Utgangspunktet for testen med sikte til venstre og blinken til høyre. (b) Siktet er sluppet over blinken og en poengsum er gitt i øverste venstre hjørne.

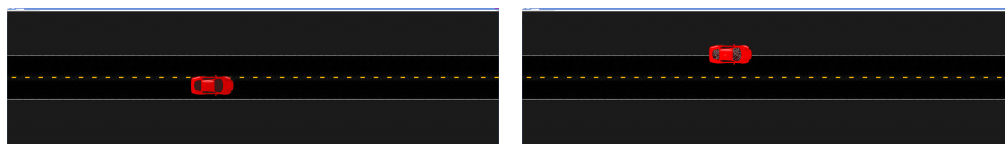
Figur 14: viser blinktestens utgangspunkt (a) og etter utførelse (b).

6.2.3 Test av hoverering

Designmessig er denne helt lik blinktesten, men i denne testen velger man siktet ved å holde markøren over siktet i to sammenhengende sekunder. For å slippe siktet på blinken må man holde siktet i ro over blinken i to sekunder. Testen har som formål å indikere hvordan det er å velge og slippe elementer ved å holde markøren over ønsket element.

6.2.4 Biltest

Formålet til denne testen er å se hvordan de ulike interaksjonsmetodene støtter det å navigere *effektivt* over større flater. Deltakeren skal flytte en bil langs en vei som strekker seg over hele skjermen, illustrert i figur 15(a). Dersom bilen flyttes utenfor veien vil dette bli registrert av systemet, og deltakeren får en tilbakemelding ved at bilen endrer utseende, illustrert i figur 15(b). Ved slutten av veien vil det totale antallet utforkjøringer registreres.



(a) Bilen flyttes langs veien

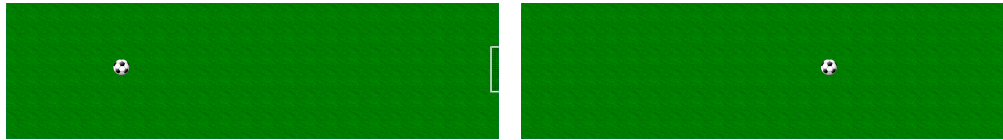
(b) Bilen kjører utfor veien

Figur 15: viser biltesten underveis (a) og dersom bilen flyttes utfor veien (b)

6.2.5 Gesttest

Selve testen er utformet som en fotballbane, illustrert i figur 16, hvor oppgaven er å flytte en fotball inn i et mål. Deltakeren manøvrerer ved å strekke en arm i den retningen man

ønsker å flytte fotballen. En slik bevegelse blir definert som en gest. Deltakeren vil bli vurdert utifra antallet gester som blir brukt for å få fotballen i målet. Hensikten med testen er å vurdere hvor god kontroll deltakerne har ved å navigere på denne måten.



(a) Utgangspunkt for testen med ballen til venstre og målet helt til høyre. (b) Ballen er flyttet to ganger mot målet.

Figur 16: viser gesttestens utgangspunkt (a) og underveis i testen (b)

6.3 Brukertest

Interaksjonsmetodene ble brukertestet med interaksjonstestene som ble presentert i seksjon 6.2. Testene ble loggført på et forhåndsdefinert skjema, vedlegg D. Utførelsen av hver test ble målt i tid, poeng og kontrollfølelse.

Hver brukertest startet med en introduksjonstekst hvor det ble presentert hva som ønskes å oppnås med testen, samt noen instruksjoner. På forhånd ble det samlet inn bakgrunnsinformasjon om deltakerne. Bakgrunnsinformasjonen inkluderte navn, alder, kjønn, yrke eller studie, eventuell tidligere bruk av gyromus og Kinect. Deltakerene ble bedt om å evaluere sin egen IT-kompetanse på en skala fra én til ti, hvor én var ingen kompetanse og ti var IT-ekspert. Egenvurderingen ble brukt for å skille mellom testpersoner med høy og lav IT-kompetanse.

Det var totalt syv deltakere i brukertesten. Deltakerne hadde varierende, men relativt mye IT-erfaring. Alle var studenter i aldersgruppen 22 - 25 år, og ingen hadde tidligere erfaring med interaksjonsmetodene.

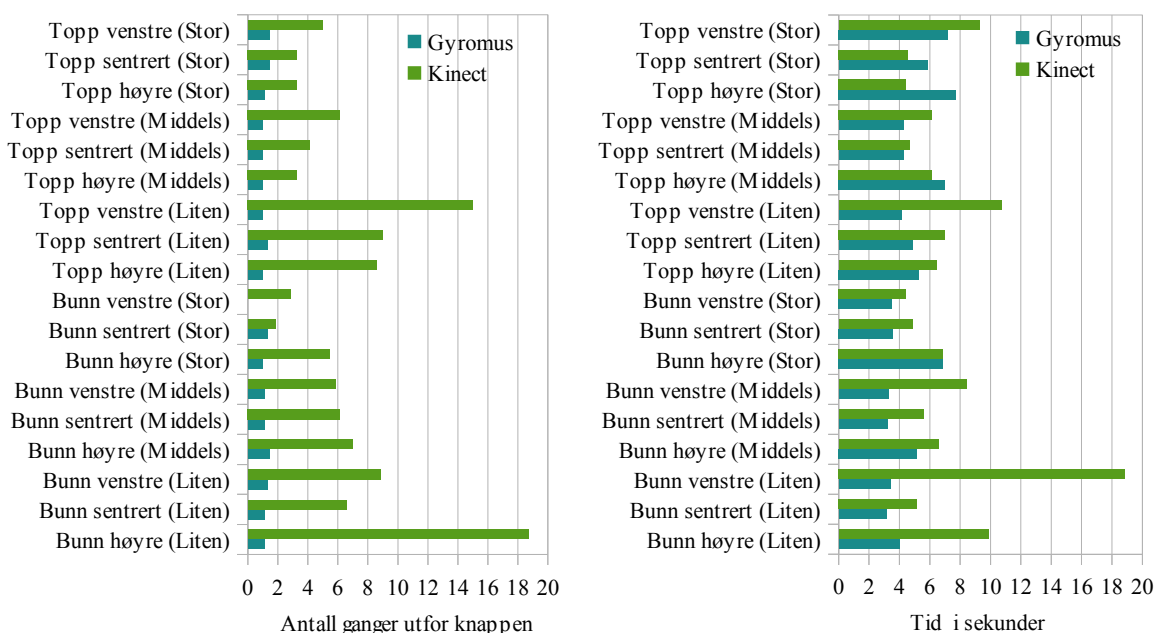
6.4 Resultat

Resultatene av brukertestene ble samlet i et regneark. Rådataene ble behandlet slik at det er mulig å se sammenhenger og ulikheter mellom deltakerne. Resultatene av dette arbeidet presenteres her som indikasjoner under hver test. Indikasjonene vil brukes videre i diskusjonen i neste seksjon.

6.4.1 Knappetest

Denne testen identifiserer interaksjonsteknologiernes *stabilitet* og *nøyaktighet*, dvs. dens evne til å gi samme resultat gjentatte ganger, samt *nøyaktighet* i form av deltakerens evne til å utføre oppgaven med få forsøk. Testen ble kjørt tre ganger for hver interaksjonsmetode, først utført med gyromus og deretter med Microsoft Kinect.

Som det fremgår av figur 17(a) havner markøren utenfor knappene vesentlig mye mer med Kinect enn med gyromus. At Kinect er mindre *presis* viser seg svært tydelig når knappene blir mindre. Interaksjonsmetoden presterte særs dårlig da den i snitt var 15 ganger utenfor knappen «Topp venstre (Liten)» og 18,7 ganger utenfor «Bunn høyre (Liten)», som illustrert i 17(a).



(a) viser hvor mange ganger markøren var utenfor (b) viser hvor lang tid det tok å «trykke» på de ulike knappene med Kinect og gyromus

Figur 17: viser antall ganger markøren var utenfor knappene i knappetesten med ulike interaksjonsteknologiene. (a) og hvor lang tid det tok å «trykke» på de ulike knappene (b)

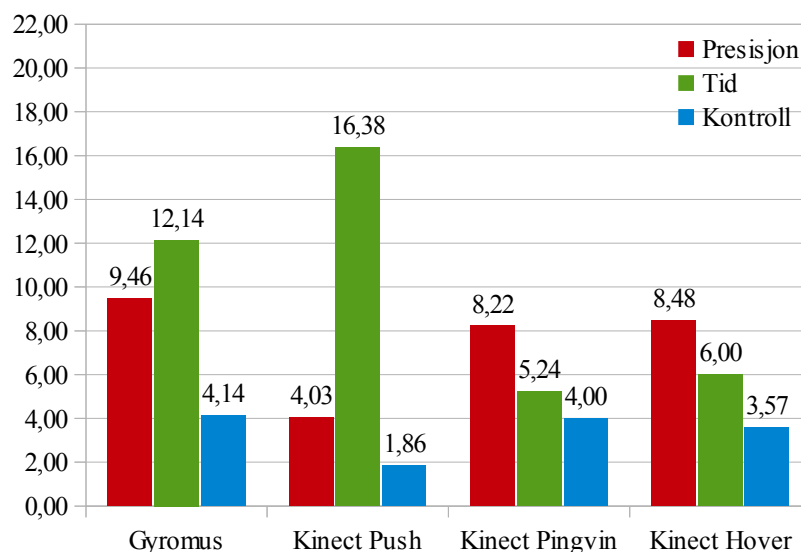
Indikasjon I1: Interaksjon med gyromus er vesentlig mer presis enn med Kinect.

Med gyromus tar det i snitt nesten dobbelt så lang tid å holde over knappen helt til høyre i forhold til med Kinect, se figur 17(b). Årsaken har trolig sammenheng med at gyromusen har en begrensning på hvor mye den beveger seg per armbevegelse og at brukeren allerede har brukt denne bevegelsen for å flytte seg fra venstre knapp til midtre knapp. Årsaken underbygges ytterligere av observasjoner i brukertestene, da det var tydelig at navigasjon over lengre distanser var vanskelig for flere av deltakerne.

Indikasjon I2: Kinect er mer effektiv over større flater enn gyromus.

6.4.2 Blinktest

Hovedfokuset til testen var å måle presisjonen til de ulike interaksjonsmetodene. Presisjonen ble målt ved at deltakerne skulle slippe et sikte på en målskive. Presisjonen til de ulike interaksjonsmetodene ble målt på en skala fra null til ti, hvor ti er i senter av målskiven.



Figur 18: viser en oversikt over resultatene fra blinktesten. De ulike interaksjonsmetodene vises i forhold til presisjon, tidsbruk og deltakernes kontrollfølelse.

Testen ble utført tre ganger med alle interaksjonsmetodene. Som vist i figur 18 tar pushmetoden lang tid, gir dårlig poengsum og lav kontrollfølelse.

Indikasjon I3: Pushmetoden er uegnet som interaksjonsmetode.

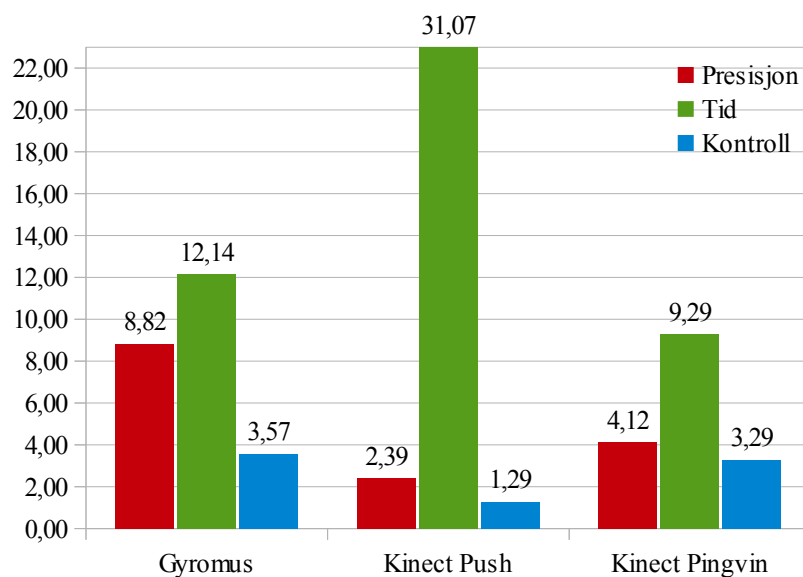
Gyromus har svært god presisjon, men bruker vesentlig lengre tid enn pingvinklikk på testene. Hvilket kan indikere at det er enklere å flytte siktet med Kinect, men kan også være et resultat av at deltakerne brukte lengre tid å finjustere med gyromus siden de har større mulighet å påvirke resultatet grunnet bedre stabilitet.

Indikasjon I4: Gyromusen har svært god presisjon, men mindre effektiv enn Kinect med pingvinklikk.

Indikasjon I5: Med unntak av presisjon, presterer pingvinklikk totalt sett meget bra.

6.4.3 Blinktest med mindre blink

Målet med denne testen var å fremprovosere forskjellene mellom gyromus, Kinect med push og Kinect med pingvinklikk. Denne provokasjonen ble oppnådd ved å ha mindre sikte og mindre blink. Testen ble utført tre ganger med alle interaksjonsmetodene. Subjektiv kontrollfølelse ble målt på en skala fra én til fem.



Figur 19: viser resultatet fra blinktesten med mindre målskive. De ulike interaksjonsmetodene vises i forhold til presisjon, tidsbruk og deltakernes kontrollfølelse.

Som illustrert i figur 19 er presisjonen ved gyromus svært høy, mens tiden som ble brukt for å plassere siktet var noe høyere enn ved pingvinklikk. Presisjonen ved pingvinklikk gikk ned fra 8,22 i den forrige blinktesten, til 4,12 i denne blinktesten. Presisjonen til gyromus

sank marginalt fra 9,46 til 8,82. Deltakerne hadde problemer med å velge siktet når det ble mindre, dette førte blant annet til at to av deltakere ikke klarte å fullføre testen med Kinect Push. For deltakerne som ikke klarte å fullføre testen med Kinect Push ble presisjon og tid ikke tatt med i resultatet, men kontrollfølelsen ble satt til én. For deltakerne som klarte å fullføre testen, økte forskjellen i presisjon betraktelig hvis en sammenligner resultatene fra figur 18 og 19.

Indikasjon I6: Kravet til presisjon øker vesentlig når elementene blir mindre.

6.4.4 Test av hoverering

Testen hadde som formål å måle den subjektive kontrollfølelsen ved velge med hoverering fremfor de andre interaksjonsmetodene. Den subjektive kontrollfølelsen ble målt på samme skala som tidligere, én til fem.

Det å velge elementer med pingvinklikk får tilnærmet lik score som når elementer velges ved å hoverere over dem. Alle aspektene, presisjon, tidsbruk og kontrollfølelse er, som vist figur 18, marginalt forskjellige mellom pingvinklikk og hoverering.

Indikasjon I7: Velging med hoverering er nesten likestilt med å velge med pingvinklikk.

6.4.5 Biltest

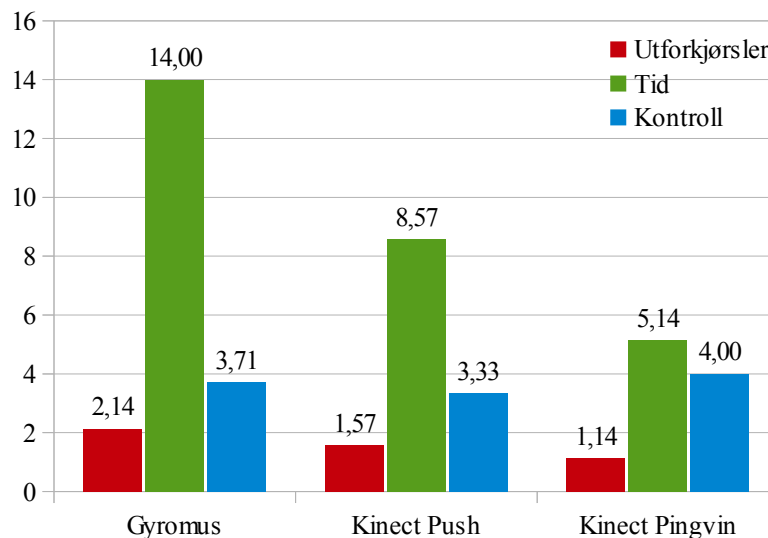
Målet med biltesten var å avdekke eventuelle forskjeller mellom de ulike interaksjonsmetodene ved flytting av elementer over lengre avstander. Testen ble utført med alle interaksjonsmetodene.

Som det kommer frem av figur 20, tok det i snitt 14 sekunder å fullføre testen med gyromus, hvilket er nesten tre ganger lengre tid enn med pingvinklikk. Gyromus presterte også dårligere enn push- og pingvinklikk på antall utforkjøringer.

Indikasjon I8: Det er svært tidkrevende å flytte et element over en lengre distanse med gyromus.

Indikasjon I9: Gyromus er mindre nøyaktig enn Kinect når elementer skal flyttes over lengre distanser.

Indikasjon I10: Kinect med pingvinklikk oppleves som den beste interaksjonsmetoden, jf. utforkjøringer, tidsbruk og kontrollfølelse.

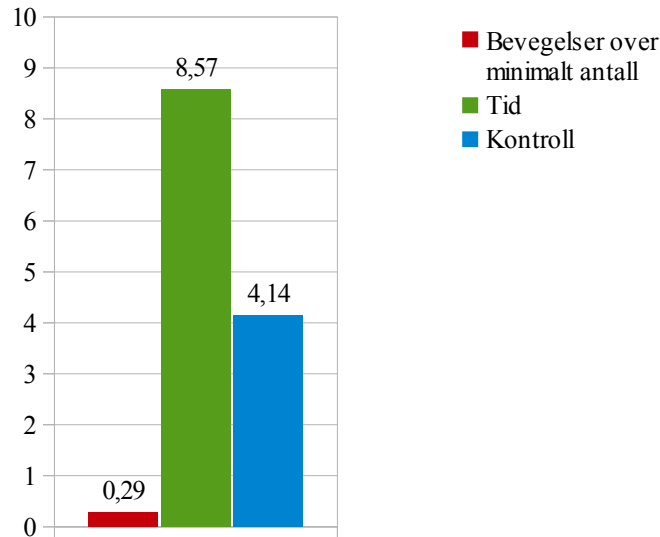


Figur 20: viser resultatet fra biltesten. De ulike interaksjonsmetodene vises i forhold til presisjon, tidsbruk og deltakernes kontrollfølelse.

6.4.6 Gesttest

Gesttestens formål var å identifisere hvor effektivt deltakerne klarte å navigere ved hjelp av gester. Testen målte den subjektive kontrollfølelsen ved gestbevegelser. Kontrollfølelsen vil her fortelle noe om forholdet mellom gestbevegelser og andre interaksjonsmetoder som bruker markørkontroll. Testen kunne løses med minimalt syv gestbevegelser. Kontrollfølelsen ble målt på samme skala som tidligere, én til fem.

De fleste av deltakerne klarte å flytte ballen i målet med minimalt antall bevegelser. Kontrollfølelsen til deltakerne var gjennomgående høy. Som det kommer frem i figur 21 tok det 8.57 sekunder å flytte ballen til målet. 8.57 sekunder vurderes som en relativt lang tid, sett i forhold til biltesten med pingvinklikk, som tok 5.15 sekunder på en lengre distanse.



Figur 21: viser resultatet fra gestttesten. Gestttesten vises i forhold til antall gester over minimalt antall, tidsbruk og deltakernes kontrollfølelse.

Indikasjon I11: Deltakerne har god kontroll ved gestbevegelser.

Indikasjon I12: Repeterende gestbevegelser tar lang tid.

6.5 Diskusjon

Gyromus oppleves som svært presis, jf. indikasjonene I1, I4 og I6. Den er derimot mindre effektiv når markøren flyttes over store avstander, jf. I2, I4, I8 og I9. For å flytte markøren over store avstander med gyromusen må man gjøre flere bevegelser med armen, dette forklarer trolig hvorfor gyromusen er mindre effektiv. Under testen kom det frem følgende kommentarer som underbygger disse indikasjonene:

Deltaker 1 om gyromus: «Litt tregt over store flater.»

Deltaker 3 om gyromus: «Føler at bevegelsen går i bue på skjermen.»

Å bevege markøren med Kinect er mindre presist da markøren hopper fra punkt til punkt, noe som vises igjen i oppgaver som stiller krav til presisjon, jf. I1, I3, I5 og I6. Det må dermed stilles krav til størrelsen på elementer som kan velges av Kinect. Årsaken til den

lave presisjonen stammer fra Kinect-kameraets oppløsning på 640x480 bildepunkter som kobles mot MIMIR sine 11520x3240 bildepunkter. Koblingen gjør at alle bevegelser blir skalert opp med et forhold på 1:18. Resultatet av skaleringen blir at det oppleves som markøren virrer.

En styrke ved bruk av Kinect er at det er effektivt å flytte elementer over store flater, jf. I4, I5 og I10. Kinect er effektiv fordi brukeren kan manøvrere markøren over hele skjermen med én armbevegelse, noe som underbygges av følgende kommentarer fra deltakerne:

Deltaker 1 om Kinect: «Enklere å flytte på musepekeren»

Deltaker 2 om Kinect: «Musen beveger seg mer effektivt over skjermen i forhold til gyromusen»

Kinect med push evalueres uegnet som interaksjonsmetode sett i lys av indikasjonene I3, I4 og I6. Særlig I3 og I4 indikerer sterkt at denne metoden ikke fungerer. Pushmetoden fungerer dårlig, fordi man bruker samme hånd for både å holde markøren i ro og for å velge elementer. Årsaken til at denne metoden feilet er at deltakerne hadde problemer med å koordinere håndens dybde og posisjon i forhold til kameraet. To av testene måtte avbrytes fordi deltakerne ikke klarte å utføre denne koordinasjonen. Det at pushmetoden evalueres som uegnet underbygges av følgende kommentarer:

Deltaker 1 om Kinect med push: «Jeg føler det er tilfeldig hvor man trykker og slipper»

Deltaker 5 om Kinect med push: «Vanskelig å trykke»

Kinect med pingvinklikk fungerer bra med tanke på å velge elementer jf. I5. En av grunnene til at denne metoden fungerer bra er trolig at man navigerer markøren med en arm og velger elementer med den andre. Koordinasjonsmessig er metoden derfor enklere å forstå og forholde seg til, noe som viser seg tydelig gjennom kommentarer fra deltakerne:

Deltaker 7 om Kinect med pingvinklikk: «Lettere enn push, nesten like bra som gyromus»

Deltaker 3 om Kinect med pingvinklikk ved sikting: «Bedre metode enn push. Man kan konsentrere seg bare om å holde hånden i ro»

Kinect med hovering fungerer bra ved velging av elementer, jf. I7. En årsak til dette er den minimale koordinasjonen som kreves for å velge elementer, og fordi man kun må holde over elementet i en kort tid. Det er imidlertid vanskelig å sammenligne resultatene, da hoverertesten kun ble utført én gang per testperson. Metoden ble verdsatt av deltakerne. Kontrollfølelsen var i snitt 3,57 og Deltaker 7 beskrev metoden som «Nesten like lett som gyromus».

Kinect med gester oppleves av brukerne som forståelig, jf. I11. Testen gir begrenset data, en konklusjon kan derfor ikke tas etter denne testen. Videre testing av denne metoden bør inneha flere varierte bevegelser for å finne styrker og svakheter. Indikasjon 12 er derfor ikke tatt i betraktning.

6.5.1 Refleksjoner rundt testene

Testen ble utført i lik rekkefølge for alle deltakere. Det kan dermed ha vært en læringseffekt som har påvirket resultatet. Testene er enkle, læringsutbyttet anses derfor ikke som vesentlig. Derimot vil uttalelser og den subjektive vurderingen trolig være påvirket, da flere av deltakerne gav en vurdering i forhold til den karakteren de hadde gitt i tidligere tester. For å eliminere denne variabelen burde det ha vært byttet om på rekkefølgen til interaksjonsmetodene. Å bytte om rekkefølgen kunne derimot gjort det vanskeligere å vurdere deltakerne mot hverandre.

Deltakerne i denne iterasjonen var syv studenter mellom 22 og 25 år, med relativt høy IT-kompetanse, utvalget anses derfor som homogent. Det kunne med fordel vært valgt et større og distribuert utvalgt for å få et mer generaliserbart resultat.

I gesttesten var fotballen plassert på linje med målet, dermed var det i realiteten bare nødvendig med en type gest for å flytte ballen til målområdet. At det kun var én gest som krevdes for å fullføre testen gjorde at tre av fire gester ikke ble benyttet av majoriteten av deltakerne. En mulig forbedring av denne testen kunne vært å plassere en hindring, f.eks. en spiller, foran målet slik at deltakeren måtte bruke flere gester. Med fordel kunne det vært flere tester for å avdekke potensielle gester som kunne blitt brukt i den endelige løsningen. Disse potensielle gestene kunne eksempelvis vært vinke-, sirkel- og sveipebevegelser.

Det var kun én interaksjonstest med hovereringsmetoden, dette gjør at resultatene har mindre validitet enn resultatene med pingvinklikkmetoden som benyttes i flere tester. Et større antall og mer varierte tester kunne med fordel vært utført for å få et klarere resultat av hovereringsmetoden. På bakgrunn av dette vil hoveringsmetoden bli forkastet som interaksjonsmetode da det ikke er gode nok data for å bekrefte prestasjonen.

6.5.2 Aktiviteten og synlighet

I seksjon 5.2 ble det stilt krav om at interaksjonsmetoden må være synlig, slik at aktiviteten foran informasjonsradiatoren kan ses av andre i rommet. Interaksjon med pingvinklikkmetoden og gyromus er begge synlige, men pingvinklikkmetoden vurderes til å være best egnet. Pingvinklikkmetoden er best egent fordi interaksjon med gyromus var mindre *effektiv* ved flytting av elementer over større flater, og flere deltakere hadde vansker med lange flytt. Det vektlegges også at man slipper å bruke en fysisk gjenstand for å interagere med pingvinklikkmetoden, noe som er mer relasjonsnært til naturlig interaksjon.

6.6 Oppsummering

I denne iterasjonen ble det først utformet tre krav i forhold til *presisjon, enkelhet og effektivitet* ved interaksjonsmetodene. Kravene ble evaluert ved at seks ulike interaksjonstester ble utviklet. Microsoft Kinect ble brukertestet med fire ulike interaksjonsmetoder: gestnavigasjon, hoverering, pushklikk og pingvinklikk. Brukertesten ble utført av syv deltakere med relativt høy IT-erfaring. Brukertesten gav et innblikk i hvordan de ulike interaksjonsmetodene ble oppfattet av deltakerne.

Som det fremgår av diskusjonen ble flere av metodene valgt bort. I forhold til interaksjon med hoverering og gester, finnes det ikke et tilstrekkelig datagrunnlag for å kunne ta en kvalifisert vurdering. Kinect med push presterte jevnt over svært dårlig, og vurderes derfor som uegnet. Gyromus demonstrerte god *presisjon*, men var lite *effektiv*. Kinect med pingvinklikk gav god *kontrollfølelse*, var *effektiv* og hadde tilstrekkelig god *presisjon*. Det naturlige aspektet ved interaksjonsmetoden har blitt vektlagt, noe som medførte at interaksjon med pingvinklikk ble valgt fremfor gyromus.

7 Iterasjon 2

I denne utviklingsiterasjonen utvikles en funksjonell og testbar prototype av Verdandi. Utviklingen innebærer at både forretningslogikk og brukergrensesnitt implementeres.

I iterasjon 1 ble pingvinklikk identifisert som den mest egnede interaksjonsmetoden, og vil derfor brukes med prototypen. Metoden var enkel og effektiv ved flytting av elementer over store flater, men krevde at elementene var av en viss størrelse. I utviklingen av prototypen må det derfor tas hensyn til størrelsen på valgbare elementer. I denne iterasjonen vil det bekreftes eller avkreftes om interaksjonsmetoden fungerer sammen med Verdandi.

Det vil i denne iterasjonen fokuseres på utvikling av grunnleggende elementer og hovedfunksjonalitet ved prototypen. Fokuset inkluderer blant annet brukerhistorie- og oppgavekort, samt flytting av oppgavekort mellom steg i prosessen. Som en del av iterasjonen vil prototypen evalueres med en brukertest, hvor målet er å avdekke brukbarhetsproblemer som brukes til å rettlede videre utvikling.

Kapittelet er strukturert ved at det først presenteres egenutviklede og teoribaserte krav samt en beskrivelse av prototypen. Deretter presenteres brukertesten og dens resultater, som videre analyseres og diskuteres. Til slutt følger det en oppsummering av iterasjonen som helhet.

7.1 Krav

Basert på resultatene fra iterasjon 1 og bakgrunnsmateriale presentert i kapittel 2 ble det utarbeidet 17 krav til Verdandi. Kravene gjenspeiler informasjonsradiatorens hovedelementer samt noen elementer som kan gi merverdi til systemet. MIMIR som videovegg gir noen begrensninger i forhold til design av grensesnittet. For å håndtere begrensningene underveis i utformingen ble det tatt hensyn til følgende punkter, som tidligere er presentert i seksjon 2.2:

- Det kan være vanskelig å holde rede på hvor markøren befinner seg.
- Det kan være vanskelig å flytte elementer som er spredt utover skjermen.
- Lange flytt kan være vanskelig og tidkrevende.

- Brukere skiller ofte ulike skjermer fra hverandre som logiske enheter, oppgaver eller idéer.
- Skjermmrammene kan være forvirrende for brukeren.

Punktene ble vurdert som begrensninger ved mediet og tas hensyn til under utviklingen. Punktene ble ikke regnet som direkte krav til Verdandi, men som retningslinjer for utformingen av grensesnittet.

Brukerhistorie- og oppgavekort er to sentrale elementer i Verdandi og siden dette er et digitalt system kan disse to elementene inneholde informasjon som ved en fysisk informasjonsradiator ville vært krevende. Ferdigstillellesprosent på brukerhistorier er et eksempel på et slikt krevende element, siden dette er et element som krever hyppig oppdatering. I tillegg bør kravene gjenspeile forslagene til innhold i brukerhistorier som beskrevet av Beck & Andres (2004) og følge Cohns mal for tekstlig innhold. Brukerhistoriene deles i oppgaver for å støtte Cohn (2004) sine to argument: flere utviklere jobber ofte med én brukerhistorie og kan derfor deles opp slik at ulike utviklere kan jobbe med det de er gode på; oppgavekort representerer arbeidsoppgaver på en bedre måte enn brukerhistorier. Videre kan brukerhistorier og oppgaver, i følge Wake (2002) og Sharp et al. (2009), notasjonsmessig skilles fra hverandre, noe som anses som en fordel for å gi et raskt overblikk. Det stilles dermed følgende krav til utformingen av brukerhistorie- og oppgavekort:

1. Brukerhistorier skal inneholde kort overskrift, tekstlig beskrivelse, prioritet, akseptanskriterie og notater.
2. Brukerhistorier skal være sortert etter prioritert rekkefølge.
3. Brukerhistorier skal ha en ferdigstillellesprosent. Prosenten skal vise antall oppgavekort som er ferdige i forhold til hvor mange som er under utvikling, eller ikke er påbegynt.
4. Brukerhistorier skal skrives på formen: som en <type bruker>, vil jeg <et mål> slik at <en grunn>.
5. Oppgavekort skal vise estimat, navn og bilde på utvikler, oppgaveidentifikator, samt tekstlig beskrivelse.
6. Oppgavekort skal være visuelt knyttet til brukerhistorier.
7. Ha ulike typer oppgavekort, eksempelvis for koding, brukergrensesnitt og bugs.

Som det fremgår i teorikapittelet, seksjon 2.1.1, er deler av informasjonsradiatorens viktigste egenskap å kunne vise progresjonen i utviklingsprosessen. For å vise progresjonen må systemet ha kolonner som beskriver de ulike stegene i utviklingen (Perry, 2008; Sharp et al., 2006). Det må også være mulig å flytte et oppgavekort fra et steg til et annet. I et digitalt system som Verdandi har man mulighet til å forhindre brukerfeil. Et eksempel på dette er å forhindre at brukerne plasserer oppgavekort på feil brukerhistorie. Kravet kan også støtte dimensjonen *utsatt for feil*, som beskrevet i seksjon 3.3.2. Følgende krav er utformet:

8. Det skal være totalt fire kolonner for å visualisere prosessen, dette inkluderer *to do*, *in progress*, *testing* og *done*.
9. Det skal være tydelig hvor i prosessen et oppgavekort befinner seg.
10. Det skal være mulig å flytte et oppgavekort fra en kolonne til en annen, men det skal ikke være mulig å flytte en oppgave mellom brukerhistorier.
11. Det skal være mulig å flytte oppgavekort frem og tilbake mellom kolonner.

Som nevnt av Perry (2008), i seksjon 2.1.4, kan burndowndiagram være en del av en informasjonsradiator og er i følge Cohn (2004) mest nyttig når de er store og synlige. Siden diagrammet, i likhet med informasjonsradiatorens kolonner, representerer progresjonen i prosjektet kan diagrammet ytterligere forsterke synligheten til prosessen. Å ha et burndowndiagram kan støtte *synlighetsdimensjonen* ved det kognitive dimensjonsrammeverket, og er således ønskelig å ha som en del av Verdandi. Burndowndiagram og generell statistikk kan også være nyttig for brukerne, da de kan bidra til å få et bedre overblikk over prosessen. Det er derfor utviklet krav om at:

12. Det skal være et burndowndiagram over nåværende iterasjon. Diagrammet skal oppdateres automatisk når en oppgave er utført, og skal alltid skal være synlig.
13. Systemet skal vise statistikk om hvordan iterasjonen ligger an, dette innebærer å vise antall gjenværende dager og brukerhistoriepoeng.

Som nevnt i teorikapittelet, i seksjon 2.1.6, blir det i Azizyan et al. (2011) fremmet et behov om å integrere andre systemer med informasjonsradiatoren. Dubakov & Stevens (2008) forutser også dette behovet, da det konkluderes med at systemer for smidig

prosjekthåndtering vil utvikle seg til løsninger som integrerer hele livssyklusen. På bakgrunn av denne integrasjonen og behovet, er det utformet krav om å integrere meldinger fra versjonshåndteringssystemer og generelle meldinger. Commitmeldinger kan øke synligheten til applikasjonen samt gi teamet et overblikk over siste endringer, noe som kan være aktuelt ved diskusjoner og møter. Det ble utformet følgende krav:

14. Systemet skal vise de siste kodeendringene i prosjektet.
15. Systemet skal vise generelle meldinger. Eksempel på meldinger kan være hvem som ikke er på jobb og at skal være et møte de nærmeste dagene.

På grunn av størrelsesbegrensningene gitt av interaksjonsmetoden og fordi all informasjon skal være leselig fra en viss avstand, kan det være en mulig løsning å navigere i en liste av brukerhistorier. Det stilles derfor krav til denne navigasjonen og til informasjonen som er i nåværende skjermbilde:

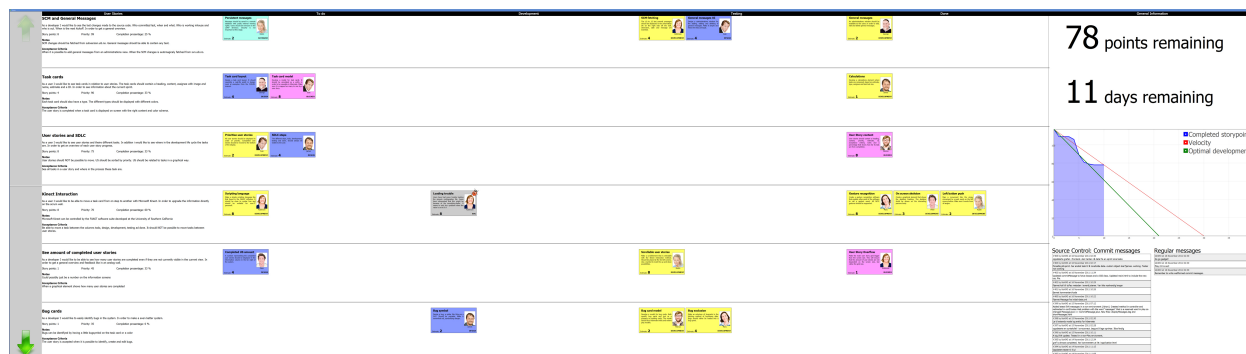
16. Det skal være mulig å navigere opp og ned i en liste av brukerhistorier. Det skal da gis tilbakemelding om hvor man er i denne listen.
17. Det skal være mulig å se hvor mange brukerhistorier som er ferdige, selv om ikke alle er synlige.

Kravene som er presentert ovenfor ble prioritert, slik at kravene ble implementert i den rekkefølgen som gav den viktigste funksjonaliteten først. De viktigste elementene, i prioritert rekkefølge, ved prototypen var: kolonner og rader, brukerhistorier og oppgaver, mulighet til å flytte oppgaver og burndowndiagrammet. Elementene ble vurdert som et minstekrav for at prototypen skulle være mulig å brukerteste, og ble derfor implementert først.

De resterende elementene, meldinger og statistikk, ble implementert sist. I tillegg til prioriteringen ble kravene estimert, slik at det var mulig å estimere hvor lang tid utviklingen av hele prototypen ville ta. På grunn av liten kjennskap til utvikling på MIMIR ble kravene estimert i forhold til hverandre med skalaen én, to, fire og åtte. Ved å estimere kravene på denne måten var det mulig å se hvor lang tid som ble brukt på de første kravene, og deretter estimere hvor mye tid som ville gå med til de resterende. En fullstendig kravkatalog er finnes i vedlegg A. Implementeringen av kravene resulterte i en prototype, som presenteres i neste seksjon.

7.2 Prototype

Prototypen som er utviklet i denne iterasjonen innehar hovedelementene ved en informasjonsradiator som presentert i seksjon 2.1. Verdandi vil bli presentert i denne seksjonen. Utformingen vises i figur 22, og en mer detaljert figur er vedlagt i vedlegg H.



Figur 22: viser Verdandi etter at utviklingen i iterasjon 2 var ferdig. Oppsettet viser utgangspunktet for brukertesten. Brukerhistorier er plassert helt til venstre, deretter vises kolonner med oppgaver. Helt til høyre er generell informasjon, burndowndiagram og meldinger.

7.2.1 Struktur

Fysiske informasjonsradiatorer gir muligheten til å se hvordan en iterasjon ligger an i forhold til en prosjektdefinert prosess. Ulike informasjonsradiatorer har, som vist av Sharp et al. (2006) ulik struktur. Verdandi benytter seg av en struktur med kolonneinndeling for oppgavekort. Kolonnene representerer, fra venstre mot høyre: hva som må gjøres, hva som er under utvikling, hva som må testes og hva som er ferdig.

Det ble tatt et designvalg om å ha både brukerhistorier og oppgaver synlig på skjermene. Brukerhistorienes oppgaver vises vertikalt i form av rader for å gi brukerne av systemet en bedre oversikt i forhold til relasjonen mellom brukerhistorier og oppgaver. Relasjonen vil i tillegg gi brukerne av systemet muligheten til å se oppgavene i en større sammenheng. Designvalget med rader og kolonner var også påvirket av utformingen til MIMIR, da skjermrammene ved mediet kan tolkes som celler eller, som ifølge Ball & North (2005), logiske enheter. Hver kolonne dekker én skjermkolonne, mens hver skjermrekke har to rader til brukerhistorier. Strukturen gjør det mulig å vise seks brukerhistorier med maksimalt åtte oppgaver i hver celle. Hvordan skjermrammene påvirker grensesnittet illustreres i figur 23.



Figur 23: illustrerer hvordan skjermrammene til MIMIR deler opp grensesnittet til Verdandi.

7.2.2 Brukerhistorier og oppgavekort

Brukerhistoriene er plassert på skjermene helt til venstre, som en liste. I tillegg til at en brukerhistorie inneholder et sett med oppgaver, inneholder de en overskrift, tekstlig beskrivelse, notater, akseptansekrav, estimat, prioritet og fullføringsprosent. Fullføringsprosent representerer et forholdstall mellom antall ferdige og ikke ferdige oppgaver i en brukerhistorie. Innholdet i en brukerhistorie er til en hver tid synlig i grensesnittet, en brukerhistorie er illustrert i figur 24.

Task cards

As a user I would like to see task cards in relation to user stories. The task cards should contain a heading, content, assignee with image and name, estimate and a ID. In order to see information about the current sprint.

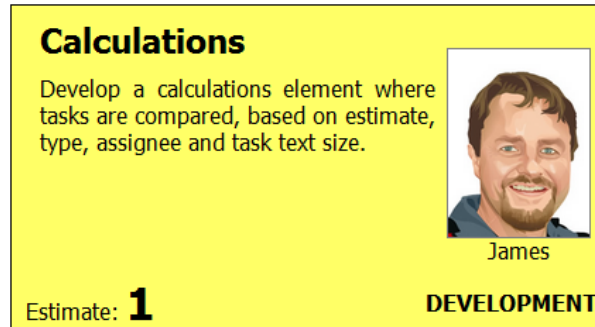
Story points: 4 Priority: 90 Completion prosentage: 33 %

Notes
Each task card should also have a type. The different types should be displayed with different colors.

Acceptance Criteria
The user story is completed when a task card is displayed on screen with the right content and color schema.

Figur 24: viser utseende til et brukerhistoriekort. Brukerhistoriene vises på skjermene helt til venstre og inneholder: overskrift, brukerhistorie, brukerhistoriepoeng, prioritet, ferdigstillelsesprosent, notater og akseptanskriterie.

Oppgaver representerer mindre deler av brukerhistorier og inneholder overskrift, tekstlig beskrivelse, estimat, type og informasjon om oppgaveansvarlig. Oppgaveansvarlig presenteres med navn og bilde. Ved å benytte bilde som en del av denne informasjonen vil dette synliggjøre hvem som er oppgaveansvarlig på en bedre måte enn med kun tekst, dette illustreres i figur 25.



Figur 25: viser et oppgavekort med overskrift, tekstlig beskrivelse, estimat, type og informasjon om oppgaveansvarlig.

For å gi brukerne av systemet raskt overblikk over hvilke type utviklingsoppgaver som er i sprinten er det benyttet fargenotasjon på oppgavekortene. Fargekodenes betydning er som følger: utviklingsoppgaver er gule, databaseoppgaver er lysgrønn, oppgaver som blokkerer for videre utvikling³⁸ er lilla, designoppgaver er blå og bugs er grå og hadde i tillegg et ikon som understreker betydningen til kortet.

7.2.3 Generell informasjon

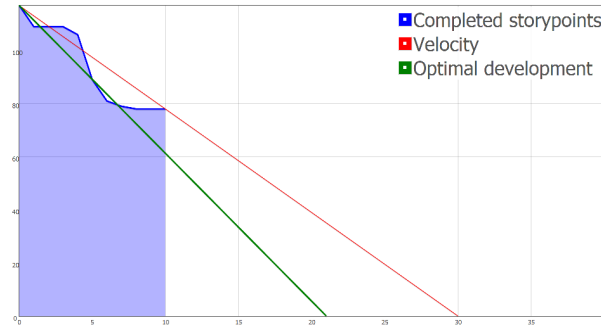
Skjermen øverst til høyre blir benyttet til generell informasjon. Blant denne informasjonen er gjenstående dager og brukerhistoriepoeng i sprinten. Informasjonen blir oppdatert automatisk for å gi brukerne en umiddelbar tilbakemelding om prosessen.

7.2.4 Burndowndiagram

Burndowndiagrammet er opprettet på bakgrunn av beskrivelsen presentert i seksjon 2.1.4. Den horisontale aksene representerer antall dager og den vertikale aksene representerer brukerhistoriepoeng. Som illustrert i figur 26 inneholder diagrammet tre grafer:

- Blå graf viser antall gjenværende brukerhistoriepoeng i sprinten.
- Grønn graf viser optimal utvikling, det vil si hvordan utviklingen bør være for å nå det initielle estimatet.
- Rød graf viser velisiteten til prosjektet og indikerer når prosjektet vil bli ferdig, basert på hvor mange brukerhistoriepoeng som blir fullført per dag.

³⁸Eng.: blocker



Figur 26: illustrerer burndowndiagrammet med tre grafer. Rød representerer velositet, grønn representerer optimal utvikling og blå representerer utført arbeid.

Både gjenværende brukerhistoriepoeng og velositet oppdateres automatisk når et oppgavekort blir flyttet til den siste kolonnen.

7.2.5 Kildekontroll- og informasjonsmeldinger

Nederste høyre skjerm viser de siste meldingene som kommer fra kildehåndteringssystemet. Meldingene fører til en synliggjøring av endringene i kodebasen, og kan bidra til å holde teamet oppdatert. Det er i tillegg mulig å publisere generelle meldinger på skjermen, som kan være alt fra påminnelser om viktige milepæler til meldinger om ferieavvikling eller møter. De to meldingstypene skilles fra hverandre i to lister.

7.2.6 Interaksjon

Pingvinklikk ble vurdert som den beste interaksjonsmetoden i første iterasjon, men på grunn av problemer med presisjonen til Kinect er det utført noen justeringer. I hovedsak innebar justeringene å sette en forsinkelse på sporingen av håndflaten, noe som resulterte i at markøren ble mer presis.

Robertson et al. (2005) og Ni et al. (2006) skriver at det er vanskelig å identifisere hvor markøren befinner seg på videovegger. På bakgrunn av dette ble markøren i prototypen forstørret, slik at den er stor og synlig.

7.3 Brukertest

Prototypen som er presentert ble brukertestet i et kontrollert miljø, beskrevet i seksjon 4.2. Brukertestene ble utført med enkeltindivider med en antatt kjennskap til brukerhistorier og informasjonsradiatorer sin rolle i smidige metoder. I denne seksjonen beskrives brukertestens deltakere og hvordan testen ble utført.

7.3.1 Deltakere

Det var totalt syv deltakere mellom 22-34 år, tre kvinner og fire menn. Deltakerne ble valgt på bakgrunn av at de tar masterkurset om «avanserte emner i programvareutvikling» ved Institutt for informasjons- og medievitenskap på Universitetet i Bergen. I kurset benyttet deltakerne informasjonsradiatorer og brukerhistorier i egne prosjekt. Gruppen vurderes til å ha noe lavere kunnskapsnivå enn sluttbrukerne, men vurderes likevel som representative for målgruppen til Verdandi.

7.3.2 Introduksjon og intervju

Før intervjuet startet ble en introduksjonstekst lest til deltakeren. Oppsummert inneholdt denne teksten presisering av formål, kort hva som skulle brukertestes, forløpet til testen og informasjon om videoopptak av testen. Introduksjon og spørsmål ble skrevet ned som en intervjuguide i vedlegg E.

Videre ble deltakeren bedt om å fylle ut et personaliaskjema med punktene: navn, alder, kjønn, yrke/utdanning, statsborgerskap og prosjektgruppen de var en del av i kurset «avanserte emner i programutvikling». Deltakerne ble bedt om å oppgi prosjektgruppen slik at det var mulig å se på gruppens informasjonsradiator hvis noe var uklart i forhold til svarene i intervjuet. Etter at personaliaskjemaet var utfyllt fortsatte intervjuet hvor målet var å avklare deltakernes kjennskap til informasjonsradiatorer, burndowndiagrammer, brukerhistorie- og oppgavekort. Intervjuet hadde tre hovedspørsmål som reflekterte disse elementene, og hvert hovedspørsmål hadde oppfølgingsspørsmål for å styre intervjuet i riktig retning.

Før opplæringsfasen med Microsoft Kinect startet ble det spurt om deltakeren hadde kjennskap til teknologien fra tidligere, for å vurdere om tidligere erfaring kunne ha en påvirkning på resultatet.

7.3.3 Opplæringsfase

I opplæringsfasen ble det vektlagt at deltakeren forstod og oppnådde god kontroll på interaksjonsmetoden. Deltakeren ble vist fem ulike elementer ved interaksjonsmetoden: kalibreringsposituren, styring av markøren, velging av elementer, samt å gi og ta kontroll.

Deltakeren fikk frihet til å prøve de ulike interaksjonsaspektene i blinktesten som ble utviklet i iterasjon 1, beskrevet i seksjon 6.2.2. Blinktesten ble utført tre ganger, hvor det i den første testen ble oppfordret til å flytte siktet rundt i ulike bevegelser. Oppfordringen ble gitt for at deltakeren skulle få en bedre følelse av kontroll og forståelse for interaksjonsmetoden.

7.3.4 Test

Før systemet ble vist til deltakeren ble det lest en introduksjonstekst i denne delen av brukertesten for å definerte rammer iht. retningslinjene til Boren & Ramey (2000), som presentert i seksjon 3.2.2. I introduksjonsteksten ble det vektlagt hvordan testen ville foregå, forklaring av think-aloud, samt viktigheten av å fortelle alt deltakeren tenkte. Deltakerne ble gjort oppmerksom på at brukertesten ble filmet.

Sesjonen var guidet rundt seks hovedspørsmål, med oppfølgingsspørsmål og oppgaver. Alle testsesjonene ble transkribert og senere kategorisert i henhold til et kategoriskjema, vedlegg C.

7.4 Koding

Råmaterialet fra brukertestene var syv videoopptak, hvert med en varighet på mellom 30-50 minutter. Hver brukertest ble delt inn i tre deler: intervju, opplæring og test. Intervju- og testdelen ble transkribert. Hvert intervju ble oppsummert, slik at intervjuene ble mer lesbare og enklere å vurdere mot uttalelser i brukertesten.

Ut i fra de transkriberte brukertestene ble et kategoriskjema opprettet som beskrevet i forskningsdesign, seksjon 4.5. Kategoriskjemaet inneholdt fem hovedkategorier: innholdsproblemer, layoutproblemer, usikkerhet, forslag og interaksjonsproblemer. På grunn av at granulariteten ved hovedkategoriene var svært store, ble det utformet underkategorier hvor granulariteten er på elementnivå i systemet. Et eksempel på

en underkategori av innholdsproblemer er: *Uttalelser viser usikkerhet om hvor mange brukerhistoriepoeng og/eller antall dager som gjenstår i prosjektet.*

Kategoriseringen av mulige brukbarhetsproblemer ble utført individuelt. Deretter ble de syv transkriptene sortert og slått sammen til et dokument. Fremgangsmåten for denne prosessen er beskrevet i seksjon 4.5.

Resultatet av kodeprosessen var totalt 125 uttalelser som ble markert i forhold til kategoriskjemaet. Av alle uttalelsene gjaldt 19 innholdet, 29 utforming, 14 interaksjon og 32 forslag til forbedringer og utvidelser. 31 uttalelser gjaldt usikkerhet i forhold til kunnskap eller om deltakeren kunne bevege seg fritt. Uttalelsene var i stor grad overlappende mellom de ulike deltakerne. Deltakerne kommenterte i flere tilfeller de samme elementene flere ganger, hver av uttalelsene ble da markert.

7.5 Analyse

I analysen vil deltakerne først vurderes i forhold til kunnskapen som ble avdekket i intervjuet. Deretter vil brukbarhetsproblem som ble identifisert vurderes. Vurderingen struktureres i forhold til de ulike delene i prototypen. Brukbarhetsproblemene er utarbeidet ut fra ett eller flere funn, da mange av brukbarhetsproblemene gjelder samme elementer. Hver seksjon presenterer brukbarhetsproblem, krav til videre utvikling og mulig løsning på kravet.

7.5.1 Deltakernes kunnskap

Som en del av intervjuet ble deltakerne bedt om å beskrive hvordan informasjonsradiatoren deres var strukturert og hvordan de brukte den. Kunnskap om informasjonsradiatorer, brukerhistorie- og oppgavekort, og burndowndiagram ble kartlagt. I denne seksjonen presenteres deltakernes kjennskap til de ulike begrepene.

Alle deltakerne hadde brukt en informasjonsradiator med brukerhistoriekort. Ingen av deltakerne har brukt oppgavekort direkte, men det kom frem at noen av brukerhistoriene ble formulert som oppgaver. Tre deltakere hadde ikke hørt om oppgavekort tidligere.

Deltakernes kjennskap til burndowndiagram var svært varierende. Av totalt syv deltakere var det tre deltakere som hadde brukt burndowndiagram tidligere, to deltakere som hevdet

å ha teoretisk kjennskap til begrepet, én deltaker som hadde brukt diagrammet på feil måte og én deltaker som det ikke er mulig å si om har kjennskap til begrepet eller ikke. Totalt fire av syv deltakere har dermed ikke brukt et burndowndiagram tidligere. Ingen av deltakerne hadde brukt Microsoft Kinect, men de fleste hadde hørt om teknologien tidligere.

Deltakerne i brukertesten deltok i tre ulike prosjektgrupper i kurset ved Universitetet i Bergen: *Pirates*, *Goblin* og *Blue*. Prosjektgruppene og deres informasjonsradiator presenteres her.

Pirates: Utviklet en bokhandel på nett som de ga navnet «Blackbooks». Det var ingen kunde eller produkteier, dette førte til at de lagde alle brukerhistorier selv. De tok utgangspunkt i en imaginær kundes behov.

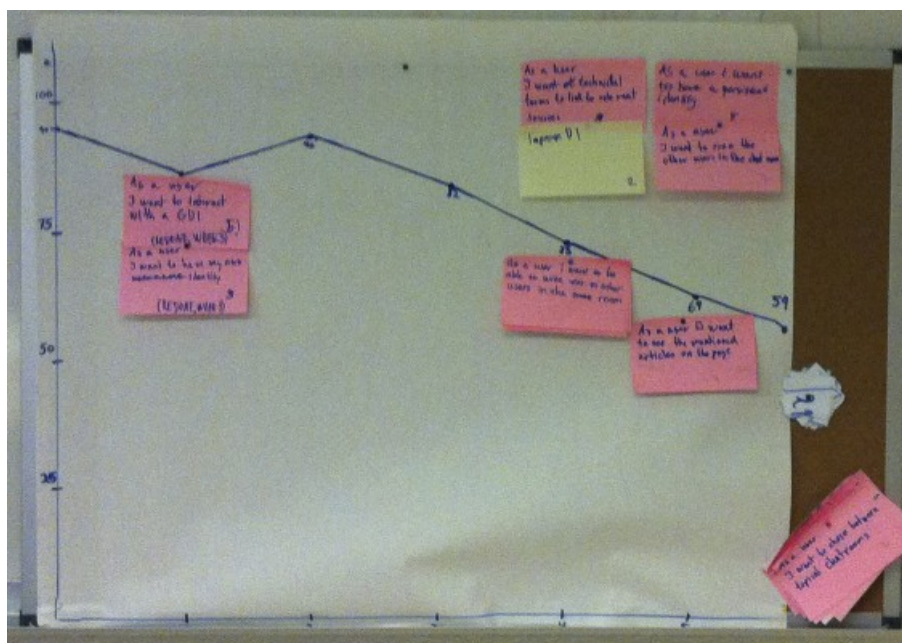
Informasjonsradiatoren ble laget på en korktavle, illustrert i figur 27. Korktavlen var, i likhet med Verdandi, delt opp i fire kolonner: «to do», «in progress» og «testing» og «done». Det ble brukt knappenåler for å lage et skille mellom kolonnene. De fire kolonneoverskriftene ble skrevet på gule postit-lapper som hang over sin respektive kolonne. Brukerhistorier ble skrevet på postit-lapper og ble hengt på tavlen. Det ble ikke brukt ulike farger for å skille mellom brukerhistoriene. Alle brukerhistoriene ble hengt på tavlen, og «to do» var en representasjon for alt som skulle gjøres. Brukerhistorier som stod på «in progress» var en del av den pågående iterasjonen, og de som stod på «testing» ventet på å bli testet før de ble satt på som ferdig. Det var ingen annen informasjon på tavlen utover postit-lappene som representerte brukerhistorier.



Figur 27: viser gruppen Pirates sin informasjonsradiator. Radiatoren er inndelt i kolonner med «to do», «in progress», «testing» og «done». De rosa postit-lappene representerer oppgaver og brukerhistorier.

Goblins: Utviklet et kategoribasert chatterom hvor nøkkelord kunne tagges. Informasjonsradiatoren var delt på to sider av en korktavle. Den ene siden var alt som skulle gjøres i prosjektet. Den andre siden viste nåværende iterasjon. Iterasjonen ble representert som et burndowndiagram, hvor ferdige brukerhistorier ble plassert som en del av grafen i diagrammet.

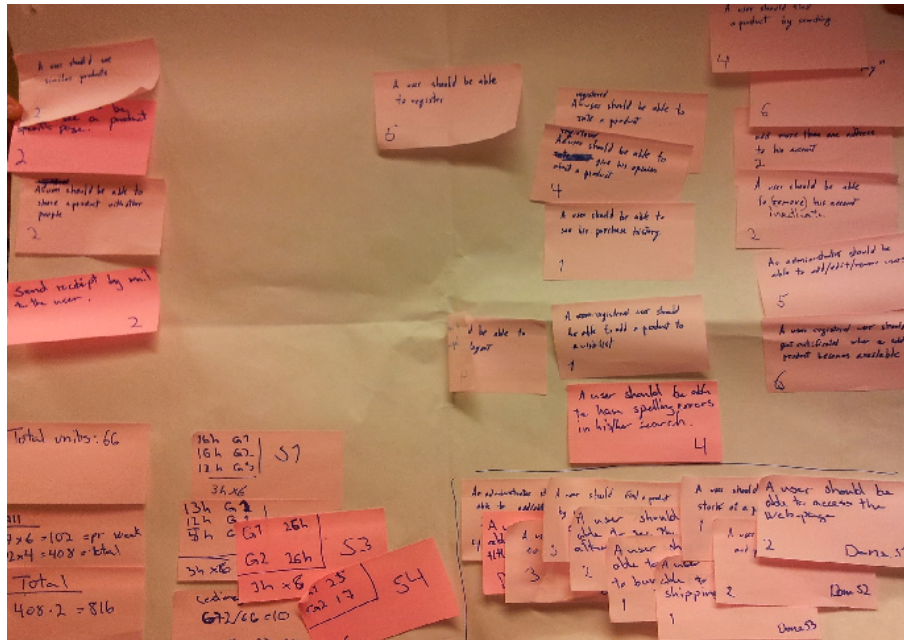
Brugerhistorier som var under utvikling ble plassert over grafen, mens brukerhistoriene som skulle gjøres ble plassert i bunke nederst til høyre på tavlen. Goblins skrev brukerhistorier på postit-lapper, og hadde i tillegg til brukerhistorien et estimat av arbeidsmengden. Det ble ikke brukt fargenotasjon for å skille ulike kort. Informasjonsradiatoren til Goblins vises i figur 28. Gruppen jobbet distribuert og brukte også et delt regneark for å dele informasjonen.



Figur 28: viser gruppen Goblins sin informasjonsradiator. Informasjonsradiatoren er utformet som et burndowndiagram hvor postit-lappene på grafen representerer ferdige brukerhistorier. Lappene øverst til høyre er under utvikling, mens lappene nederst til høyre skal gjøres.

Blue: Utviklet en nettbutikk som solgte diverse varer. Informasjonsradiatoren var delt opp i kolonner, der brukerhistoriekort var prioritert fra venstre til høyre. Brukerhistoriekortene som var i kolonne en, hadde en høyere prioritet enn de i kolonne to, og så videre. Informasjonsradiatoren hadde en rute der brukerhistorier som var ferdige ble plassert. Foruten denne ruten var det ingen notasjon som viste hvor langt i prosessen

hver enkelt brukerhistorie hadde kommet. Blue brukte ikke farger for å gi ekstra notasjon til informasjonsradiatoren, og er illustrert i figur 29. For å dele informasjonsradiatoren digitalt, tok gruppen bilder underveis.



Figur 29: viser gruppen Blue sin informasjonsradiator. Postit-lappene viser alle brukerhistoriene i prosjektet. Brukerhistoriene er prioritert fra venstre til høyre. Ferdige brukerhistorier plasseres i en rute nederst til høyre. Postit-lappene nederst til venstre er ikke brukerhistorier, men lapper som inneholder informasjon som velisiteten og totalt antall brukerhistorier.

7.5.2 Informasjon, burndowndiagram og meldinger

I denne seksjonen presenteres brukbarhetsproblemer som omhandler informasjonsskjermene, dvs. generell sprintinformasjon, burndowndiagram og informasjons- og commit-meldinger.

Det er ingen informasjon om informasjonsradiatoren viser en spesifikk iterasjon eller prosjektet som helhet. Det ble identifisert at minst tre deltakere var usikre på hvilket nivå informasjonsradiatoren viste, som resulterte i følgende krav til vider utvikling:

1. En bruker skal umiddelbart vite hvilket nivå (prosjekt, release eller iterasjon) man befinner seg på.

En mulig løsning for å tydeliggjøre nivået kan være å ha denne informasjonen tilgjengelig i grensesnittet.

Flere deltakere hadde problemer med å tolke informasjon fra burndowndiagrammet. Burndowndiagrammet viser en optimal utviklingshastighet som må til for å nå ferdigstillelsesdato, kalt «optimal development». Minst to deltakere hadde problemer med å forstå dette begrepet. Et annet problem som ble identifisert var at minst to deltakere ikke forstod at horisontalaksen i diagrammet viste dager. Følgende krav ble utarbeidet:

2. En bruker skal kunne se hva som er korrekt tidsenhet.
3. Begrepet «Optimal development» må endres.

Mulige løsninger til disse kravene kan være å legge til en tidsenhet på burndowndiagrammet, samt endre begrepet «optimal development» til «estimert velositet».

Systemet viser hvor mange brukerhistoriepoeng og dager som gjenstår i en iterasjon. Likevel var det minst to deltakere som ikke klarte å gjengi hvor mange brukerhistoriepoeng som gjenstod i iterasjonen. Videre var det en deltaker som ikke la merke til gjenstående dager. Som et resultat av funnene ble det identifisert et mulig behov for å vise informasjon om den opprinnelige mengden brukerhistoriepoeng og dager. Følgende krav ble utformet:

4. Grensesnittet bør vise hvor mange brukerhistoriepoeng det var opprinnelig.
5. Grensesnittet bør vise hvor mange dager det var opprinnelig og hvor mange dager som gjenstår i nåværende sprint.

Kravene kan løses ved å vise opprinnelig og gjenstående brukerhistoriepoeng og dager på informasjonsskjermene.

Generelle- og commitmeldinger vises på skjermen nederst til høyre. Generelle meldinger er beskrevet som «Regular messages». Uttalelser viser at dette er et lite presist begrep. Det viste seg også at selve teksten i meldingene forvirret deltakerne. Følgende krav ble derfor utformet:

6. Regular messages bør endres til en mer forklarende tekst.

7. Innholdet til regular messages bør endres.

En alternativ tekstlig forklaring som «Information messages» vil trolig være mer presist. Innholdet i meldingene bør endres til mer reelle informasjonsmeldinger.

7.5.3 Brukerhistorie- og oppgavekort

Flere deltakere var usikre på hvor mange brukerhistorier som var tilgjengelig i grensesnittet, dette kan ha opphav i flere designvalg. Skjermene viser ikke alle brukerhistorier, man må navigere ned hvis det er mer enn seks brukerhistorier i systemet. Følgende krav ble utformet:

8. Grensesnittet bør vise hvor mange brukerhistorier det er i nåværende periode.

Mulig løsning på kravet kan være å vise informasjon om hvor mange brukerhistorier som er i nåværende periode på skjermen øverst til høyre, samt ha en tallindikator på navigeringsknappene eller på et tilsvarende sted.

Minst fire deltakere syntes at brukerhistoriekortene var uoversiktlige. I tillegg oppdaget én av deltakerne en feil med mengden poeng på brukerhistorier og oppgaver, ved at brukerhistoriekortenes poengsum ikke stemte overens med totalsummen fra poengene til oppgavekortene. Følgende krav ble derfor utviklet:

9. Endre brukerhistoriekortene slik at informasjonen blir mer oversiktlig.

10. Tydeliggjøre prioriteten til brukerhistorier.

11. Brukerhistoriekortenes poengsum skal stemme med totalsummen av oppgavekortenes poeng.

En mulig løsning på dette vil være å endre utseende slik at brukerhistoriekortene ikke viser all informasjon som tekst, men at de heller viser deler av informasjonen grafisk ved ikoner eller grafer.

Verdandi viser brukerhistorier og deres tilhørende oppgaver. Kobling mellom brukerhistorier og oppgaver var ikke kjent for deltakerne i brukertesten, da de fleste hadde

benyttet begge deler uten å anse dem som forskjellige artefakter. Løsningen med radbasert inndeling ble ikke godt nok oppfattet og totalt seks av syv deltakere ble forklart koblingen mellom brukerhistoriekort og oppgavekort. Et krav ble derfor opprettet:

12. Tydeliggjøre koblingen mellom brukerhistorier og oppgaver.

For å gjøre koblingen tydelig kan det brukes radene avskilles med to forskjellige fargenyanser. I tillegg kan hver kolonne ha en mer beskrivende tekst som tydeliggjør innholdet, eksempelvis «Tasks to do» i stedet for «To do».

I prototypen som ble brukertestet ble det benyttet fargenotasjon på ulike oppgaver, samt et ikon som tilleggsnotasjon på bugkort. Tilleggsnotasjonen gjorde flere deltakere usikre på oppgavenes betydning. Det ble utformet følgende krav:

13. Ha lik notasjon mellom oppgavekort.
14. Oppgavekortenes tekstlige typebeskrivelse kan synliggjøres mer.

En mulig løsning kan være å ha lik notasjon mellom oppgavekortene, dette kan gjøres ved å ha ikon, farge og en tekstlig beskrivelse på alle oppgaver.

En ferdigstillellesprosent på brukerhistoriekortene viser hvor mange av oppgavene som er ferdig. Prosenten er den eneste indikasjonen på at en brukerhistorie er ferdig, utenom oppgavekortenes plassering. I brukertesten ble det gitt forslag om å markere ferdige brukerhistorier slik at disse var mer synlig. Det ble derfor utformet følgende krav:

15. Grafisk fremstille at en brukerhistorie er ferdig.
16. Flytte ferdige brukerhistorier til bunnen av listen.

En mulig løsning kan være å markere brukerhistorier som er ferdige med en egen farge og/eller flytte ferdige historier til bunnen av listen med en animasjon.

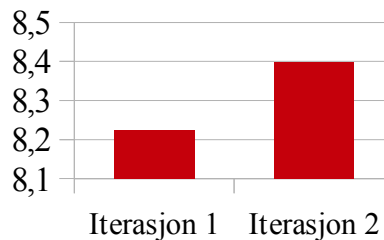
Oppgavekortene kan legges hvor som helst i mengden av oppgavekort som ligger innad i en kolonne. Rekkefølgen har ingen betydning og var et forvirrende element for minst én av deltakerne, derav følgende krav:

17. Et oppgavekort skal sorteres når den slippes fra en kolonne til en annen.

En løsning på dette kan være å sortere kortene etter rekkefølgen de ble sluppet i kolonnen.

7.5.4 Interaksjon

For å få en indikasjon på hvordan rekalkibreringen av Microsoft Kinect, beskrevet i seksjon 7.2.6, fungerte ble resultatene av blinktest notert. Testen hadde hensikt om å måle presisjonen til interaksjonsmetoden. Presisjonen ble målt ved å la deltakerne plassere et sikte, så sentrert som mulig, på en målskive. Presisjonen ble målt mellom null og ti, hvor ti var best. Gjennomsnittet ble 8.4 i denne iterasjonen som er noe bedre enn i iterasjon 1. Tid ble ikke registrert da hensikten for testen var å bli vant med interaksjonsmetoden. En sammenligning av iterasjon 1 og 2, vises grafisk i figur 30.



Figur 30: viser at gjennomsnittlig presisjon til blinktesten i iterasjon 2 har steget i forhold til iterasjon 1.

Navigasjonsmetoden for å flytte seg i mengden med brukerhistorier viste seg i brukertesten å være dårlig. Ingen av deltakerne hadde god kontroll på denne navigasjonsmetoden og tre deltakere foreslo å navigere ved hjelp av gester. Det ble også foreslått at knappene burde indikere hvor mange brukerhistorier som var over og under nåværende utsnitt. Én deltaker etterlyste en tilbakemelding når man nådde slutten av listen. Det ble utformet følgende krav:

18. Utvikle en navigasjonsmetode hvor man enkelt kan navigere blant brukerhistoriene.
19. Indikator som viser at det er flere brukerhistorier lenger ned/opp.
20. Vise grafisk/hindre når man er kommet til slutten av listen med brukerhistorier.

Navigasjonsmetoden kan implementeres ved hjelp av gester, piler som er sentrert på skjermen eller egen meny. Det må vises en oversikt over hvor mange brukerhistorier

som befinner seg over og under nåværende utsnitt, oversikten kan plasseres på informasjonskjermene eller som et eget element. Systemet bør forhindre videre navigasjon når man har nådd den nederste brukerhistorien.

7.5.5 Utvidelser

I brukertestene kom det fram en rekke forslag til utvidelser av systemet. Forslagene kom både uoppfordret fra deltakerne og etter å ha bli spurt direkte. Forslagene ble vurdert etter følgende kriterier:

- Vil det forbedre hovedfunksjonaliten til systemet?
- Hvor krevende er forslaget å implementere?
- Er forslaget teknisk mulig å implementere med det utstyret som er tilgjengelig?
- Er forslaget innenfor rammene til prosjektet?

Ut i fra disse kriteriene vil følgende forslag til utvidelser bli tatt med i neste iterasjon:

21. Systemet skal vise hvem som jobber på prosjektet.
22. Systemet skal vise kompetansen til de ulike utviklerne.
23. Commitmeldingene bør generaliseres.
24. Implementere en funksjonalitet for å se hvem som har flyttet oppgavekortene.

Følgende forslag tas ikke med i iterasjon 3:

- Opplæringsfunksjon.
- Kombinere commitmeldinger med spesifikke oppgavekort.
- Mulighet til å lage brukerhistorier.
- Mulighet til å flytte flere oppgavekort om gangen.
- Koble commitmeldinger med brukere.
- En bruker skal kunne se oversikt over prosjektet, release og pågående iterasjon.
- En måte å se hele prosjektinformasjonen.
- Vise hvor mange poeng hver utvikler jobber med.
- Muligheten til å zoome på skjermen.

7.5.6 Kravkatalog

Kravende ble satt sammen til en kravkatalog som inneholder 24 identifiserte krav. Prioriteringen av krav *til denne* iterasjonen ble utført ved at kravene ble rangert fra 0 til 100. Under implementasjonen av kravene ble det identifisert at å prioritere kravene fra 0 til 100 ikke var hensiktsmessig i denne sammenheng. Kravene som er presentert over, ble derfor prioritert som lav, middels eller høy. Prioriteringen ble utført på bakgrunn av hvor mange deltakere som hadde identifisert samme eller nært relaterte problem, samt en vurdering av hvor viktig kravet er for hovedfunksjonalitet til system. En fullstendig kravkatalog er vedlagt i vedlegg B, og vil bli brukt som bakgrunnsmateriale til utviklingen i neste iterasjon.

7.6 Diskusjon

Funnene i analysen avdekket totalt 24 krav til videre utvikling. Diskusjonen vil følge samme struktur som analysen, og det vil bli gjort rede for mulige årsaker og løsninger der dette er mulig.

7.6.1 Deltakernes kunnskap

Det generelle kunnskapsnivået blant deltakerne var lavere enn antatt, men stort sett viste de en tilstrekkelig forståelse for de ulike artefaktene i smidige metoder. Alle deltakerne hadde brukt informasjonsradiatorer tidligere, og hadde kjennskap til brukerhistorier. Ingen hadde brukt en kombinasjon av brukerhistorie- og oppgavekort tidligere. De tre prosjektgruppene hadde svært ulike informasjonsradiatorer. To av prosjektgruppene sine informasjonsradiatorer var svært langt unna strukturen i Verdandi, dette gjorde at brukertestene hadde et dårligere utgangspunkt enn forventet, noe som gjenspeiler seg i resultatene.

7.6.2 Informasjon, burndowndiagram og meldinger

Sprintinformasjonen var utviklet på bakgrunn av egne vurderinger om hva som kunne gi et raskt overblikk over fremgangen i sprinten. Det fremgikk av brukertestene at deltakerne synes informasjonen var nyttig for å få oversikt og ga dermed verdi til systemet. Brukertestene avslørte derimot at denne informasjonen kunne utvides, da flere

av deltakerne kom med forslag som vil styrke denne delen av systemet. Flere elementer, som sprintnummer og antall brukerhistorier, vil derfor bli implementert.

Enkelte av deltakerne hadde ikke brukt burndowndiagrammer i praksis, men hadde derimot teoretisk kjennskap til artefaktet. Denne mangelen på kjennskap kan ha skapt problemer i enkelte deler av testen, da spesielt ved spørsmål om prosjektets fremdrift. Ordlyden «optimal development» gjorde at flere av deltakerne ble usikre på hva denne grafen betydde. Tekstene på grafene bør derfor endres til begrep som brukes i teorien til Scrum.

Brukertesten avdekket ikke noen kritiske brukbarhetsproblemer som omhandlet commit- eller generelle meldinger. Kunnskapen til deltakerne hva angår kildehånderingsverktøy var varierende, det er derfor vanskelig å trekke noen slutninger om nytteverdien til meldingene. Det viser seg også at det tekstlige innholdet til meldingene kan påvirke hva meldingene tolkes til å være, hvilket må tas hensyn til i fremtidige tester.

7.6.3 Brukerhistorie- og oppgavekort

Brukerhistoriene ble av flere deltakere karakterisert som lite oversiktlige med alt for mye tekst. At brukerhistoriene var uoversiktlige førte til at deltakerne hadde problemer med å få oversikt over innholdet til hver enkel brukerhistorie. Særlig hadde de problemer med å tolke prioriteten til brukerhistoriene. Krav ble utformet for å utbedre brukerhistoriene, med mulig løsning om å trekke prioritet og ferdigstillellesprosent ut i grafiske elementer. Ved å benytte grafiske elementer vil det tekstlige innholdet bli mindre, og samtidig mer oversiktlig.

Ingen av deltakerne hadde benyttet typenotasjon på oppgavekort tidligere. To av syv identifiserte typenotasjonen som mindre forståelig når de ble bedt om å forklare hva de ulike fargene betydde. Hvert oppgavekort har, i tillegg til fargenotasjonen, en tekstlig beskrivelse av oppgavetyper. Ekstranotasjonen, typeikonet, på bugkort forvirret flere deltakere, da ekstranotasjonen ikke var synlig på andre oppgavetyper. At flere deltakere hadde problemer med typenotasjonen kan bety at den bør være mer konsekvent.

Fem av syv deltakere visste ikke om forskjellen på brukerhistoriekort og oppgavekort, hvilket gav en mindre bra forutsetning for inndelingen. Denne manglende kunnskapen gjenspeiles i resultatet av brukertesten, da seks av syv ikke forstod koblingen mellom

brukerhistorie- og oppgavekort.

Fem av de seks deltakerne som i utgangspunktet ikke forstod koblingen, gjorde dette når koblingen ble forklart av observatør. Det var bred enighet blant deltakerne at koblingen burde vært tydeligere, noe som indikerer at inndelingen må visuelt forbedres i neste iterasjon.

7.6.4 Interaksjon

Interaksjonsmetoden med Microsoft Kinect, hvor man styrer markøren ved hjelp av håndflaten, ble i første iterasjon referert til som noe ustabil. I denne iterasjonen ble nye innstillinger i forhold til kontroll over markøren implementert. De nye innstillingene medførte at ingen av deltakerne opplevde interaksjonen som et stort problem, selv om noen av deltakerne påpekte at det var litt uønsket bevegelse på markøren. En sammenligning mellom iterasjon 1 og iterasjon 2 viser en økning på 0.19 poeng i snitt på blinktesten. Tatt i betraktning at deltakerne i første iterasjon hadde en vesentlig bedre opplæring kan resultatet fra blinktesten anses som en indikasjon på at optimaliseringen var vellykket. En av hovedgrunnene til at interaksjonsmetoden fungerte var at alle elementer er av tilstrekkelig størrelse, og da behøver man ikke en veldig presis styring av markøren. Interaksjonsmetoden bekreftes således som egnet til Verdandi.

Navigasjonen mellom brukerhistorier var dårlig implementert, noe som viste seg svært tydelig i alle brukertestene. Ingen av deltakerne hadde kontroll over navigasjonsmetoden og utbedringer må foretas. Det ble identifisert at flere av deltakerne hadde problemer med å flytte musepekeren på skjermene helt til venstre, noe som indikerer at det ikke er det beste området for navigasjon. Et annet interessant funn er at noen av deltakerne ønsket å navigere ved hjelp av gester, hvilket kan være en mulig løsning på problemet.

7.6.5 Refleksjoner rundt iterasjonen

En stor del av brukbarhetsproblemene kan være grunnet en begrenset erfaring med begreper blant deltakerne. Blant disse begrepene var «oppgavekort» et begrep som få deltakere hadde kjennskap til. Testgruppen var valgt basert på en antakelse at de hadde en god kjennskap til bruken av informasjonsradiatorer. Antakelsen viste seg å bare være delvis riktig.

Navigasjonsknappene var ikke godt nok implementert, noe som førte til at samtlige deltakere hadde problemer med å navigere mellom brukerhistoriene. Dårlig implementasjon av navigasjonsknappene kan ha ført til at deltakerne fikk et dårlig inntrykk av prototypen og at resultatene gjenspeiler dette.

7.7 Oppsummering

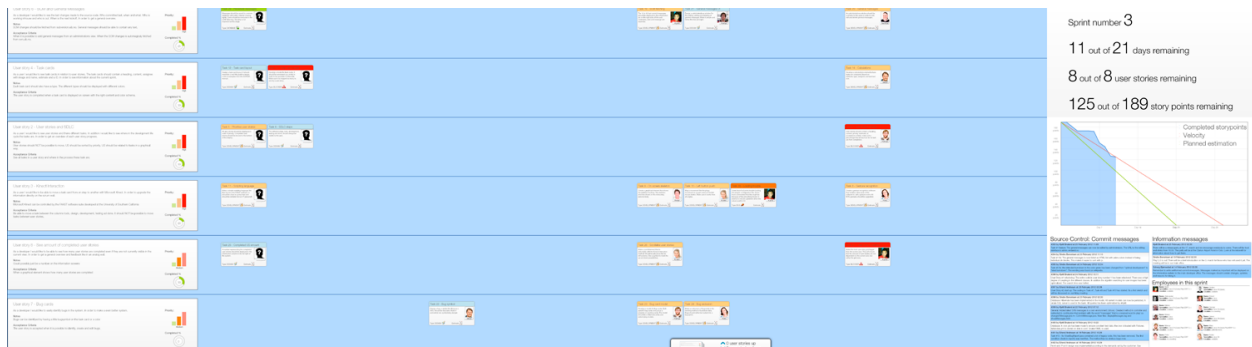
I denne iterasjonen ble det på bakgrunn av første iterasjon, begrensningene ved MIMIR, egne vurderinger og teori, opprettet 17 krav til systemet. Ut i fra kravene ble det utviklet en prototype hvor fokuset var på hovedfunksjonalitet. Prototypen inneholdt kolonner og rader, brukerhistorier og oppgaver, burndowndiagram og statistikk, samt generelle og commitmeldinger. Interaksjon med pingvinklikk ble ytterligere optimalisert og bekreftet som egnet interaksjonsmetode. Prototypen ble brukertestet av syv studenter med noe kjennskap til smidige metoder. Brukertestene ble filmet, og senere transkribert. Transkriptene ble kodet og analysert, noe som avdekket mulige brukbarhetsproblemer. Brukbarhetsproblemene og forslagene til forbedringer ble analysert, hvilket resulterte i totalt 24 nye krav til videre utvikling av Verdandi.

8 Iterasjon 3

Det fremgår av oppsummeringen i iterasjon 2 at hovedfunksjonaliteten ved systemet er implementert og at interaksjonsmetoden fungerer til formålet. Det ble identifisert 24 nye krav til systemet, som ble gitt lav, middels eller høy prioritet. Hovedfokus til denne iterasjonen er å identifisere og implementere gode løsninger til kravene. Prototypen som utvikles i denne iterasjonen presenteres i neste seksjon, og videre følger beskrivelse av brukertesten og resultatene av denne. Resultatene av brukertesten analyseres og diskuteres før iterasjonen som helhet blir oppsummert.

8.1 Prototype

Prototypen som er utviklet i denne iterasjonen har de samme hovedelementene som tidligere, men ulike aspekter er forsterket eller forbedret i henhold til resultatene fra iterasjon 2. Disse aspektene, og noen utvidelser, som er utført i denne iterasjonen presenteres i følgende seksjon. Figur 31 viser nåværende prototype. En mer detaljert figur er vedlagt i vedlegg I.



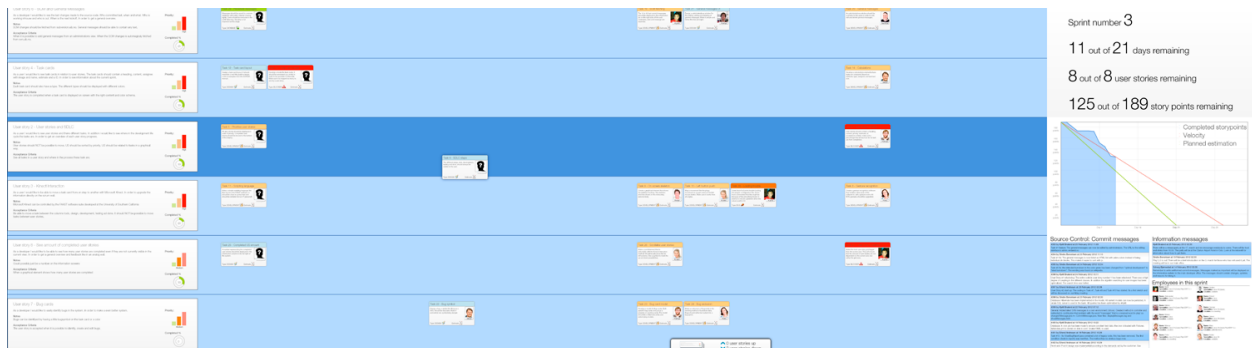
Figur 31: viser Verdandi etter at utviklingen i iterasjon 3 var ferdig. Oppsettet viser utgangspunktet for brukertesten. Plasseringen av elementer er lik som i iterasjon 2.

8.1.1 Struktur

Som tidligere består Verdandi av seks kolonner: *user story sprint backlog*, *tasks to do*, *tasks in development*, *tasks in testing*, *tasks done* og *sprint information*. Kolonneinndelingen er uendret, men overskriftene på de ulike kolonnene er endret for å tydeliggjøre hva som er i de ulike kolonnene. Overskriftene ble endret for å løse kravet om å tydeliggjøre koblingen mellom

brukerhistorier og oppgavekort. Overskriftene på oppgavekolonnene er forbedret ved at mer beskrivende tekst er lagt til. Eksempelvis er overskriften «To do» endret til «Tasks to do» og tilsvarende for resterende kolonner.

For å tydeliggjøre koblingen mellom brukerhistorier og oppgaver ytterligere blir raden hvor markøren befinner seg markert, illustrert i figur 32. For å ytterligere synliggjøre rollen til ulike elementer ble overskriften på brukerhistorier og oppgaver endret slik at de inneholder hhv. brukerhistorienummer og oppgavenummer etter de spesifikke overskriftene, med formatet «Task 1» og «User story 1».



Figur 32: viser hvordan en brukerhistorie blir uthevet når markøren holdes over raden.

8.1.2 Brukerhistorie- og oppgavekort

Brukerhistoriene er innholdsmessig like som i iterasjon 2, men for å oppfylle kravene

- «Endre brukerhistoriekortene slik at informasjonen blir mer oversiktlig» og
- «Tydeliggjøre brukerhistoriers prioritet»,

ble prioritet og ferdigstillellesprosent endret fra tekstlig til grafisk representasjon. I tillegg ble strukturen til brukerhistoriene endret slik at bredden på teksten til en brukerhistorie ble mindre. Denne strukturelle endringen ble utført for å gjøre brukerhistoriene mer oversiktlige og leselige. Den nye utformingen av brukerhistoriekort er illustrert i figur 33.

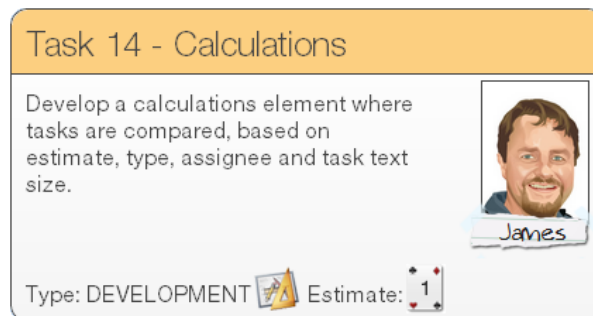
Oppgavene er, i likhet med brukerhistoriene, innholdsmessig uendret fra iterasjon 2, men utformingen av oppgavene er endret. I iterasjon 2 var hele bakgrunnen til oppgavekortet farget for å angi oppgavetype. For å gjøre teksten i oppgavekortet mer tydelig, har kortet

blitt endret slik at det kun er bakgrunnen på overskriften som er farget. De ulike fargenes betydning er også endret for å begrense mengden farger i systemet.

For å ytterligere styrke oppgavenes typenotasjon ble det implementert lik typenotasjon på den tekstlige beskrivelsen, fargen til bakgrunnen og typen til ikonene. Som illustrert i figur 34 er oppgaveestimatet endret fra en tekstlig representasjon til et ikon. Endringen ble utført for å ha et konsekvent design på nederste delen av oppgavekortene.



Figur 33: viser utseende til et brukerhistoriekort. Til venstre inneholder kortet overskrift, brukerhistorie, notater og akseptansekriterie. Til høyre vises brukerhistorien sin prioritet og ferdigstillelsesprosent som grafiske elementer.



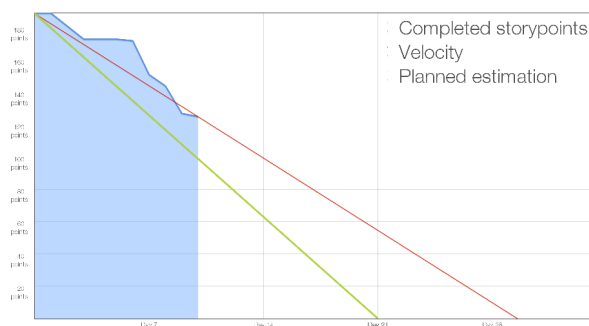
Figur 34: viser et oppgavekort. Kortet inneholder overskrift, oppgave, utvikler, type og estimat.

8.1.3 Burndowndiagram

På bakgrunn av kravene

- «En bruker skal kunne se hvilken tidsenhet som vises på burndowndiagrammet» og
- «En bruker skal kunne se når prosjektet er estimert til å være ferdig»,

ble burndowndiagrammet forbedret ved tydeligere tekster i grafene og punkter på diagrammet. Tidligere den viste den horisontale tidsaksen punkter hver femte dag uten tidsenhet, dette er forbedret ved at aksene nå viser punkter hver syvende dag med tidsenhet. Den siste dagen i en iterasjon er markert med fet skrift for å tydeliggjøre når iterasjonen skal være ferdig. Den vertikale aksene er endret tilsvarende, slik at denne nå viser gjenstående brukerhistoriepoeng med benevning, illustrert i figur 35.



Figur 35: illustrerer burndowndiagram etter endringer i iterasjon 3.

I iterasjon 2 var det flere deltakere som var usikre på hva de ulike grafene i diagrammet betydde. Tekstene til grafene ble endret for å tydeliggjøre begrepene betydning.

8.1.4 Sprintinformasjon

Med bakgrunn i forslagene til forbedringer er den generelle sprintinformasjonen utvidet. Sprintinformasjonen viser nå:

- Brukerhistorier i sprinten og hvor mange som er ferdige.
- Brukerhistoriepoeng og hvor mange som er utført.
- Dager og hvor mange som gjenstår.
- Nåværende sprintnummer.

Punkt nummer fire oppfyller kravet om systemets evne til å kommunisere at det vises en iterasjon, ikke en release eller et prosjekt. Punkt nummer to og tre ble implementert for at brukerne skal slippe å tolke denne informasjonen fra burndowndiagrammet. Antall brukerhistorier i iterasjonen ble lagt til for å gi en indikasjon om at det er flere brukerhistorier enn det som vises i utsnittet.

8.1.5 Brukerinformasjon

Oppgavekortene i forrige iterasjon var på forhånd tildelt utviklere i alle steg, til og med i *to do*. Tildelingen er nå endret slik at det er brukeren av systemet som tildeler oppgaver til seg selv. Funksjonaliteten oppnås i nåværende løsning ved at systemet, på forhånd, vet hvem som er brukeren. En videreutvikling av denne funksjonaliteten kan være å bruke ansiktsgjenkjenning med kameraet i Kinect. En slik brukerinformasjon vil også gjøre det mulig å se hvilken bruker som har oppdatert informasjonsradiatoren, og når dette er utført. Funksjonaliteten ble foreslått som en mulig forbedring av en av deltakerne i iterasjon 2.

8.1.6 Commit- og informasjonsmeldinger

Deltakere i brukertesten av den første prototypen var usikre rundt innholdet i commit- og informasjonsmeldinger. Innholdet i meldingene ble endret i denne iterasjonen, slik at innholdet fremstår som mer reellt. Tidligere var overskriften på informasjonsmeldingene «regular messages», overskriften ble endret til «information messages» for å bedre beskrive innholdet. Figur 36 viser den nye utformingen på meldingene.

The screenshot shows a web interface with two main panes. The left pane, titled "Source Control: Commit messages", lists several commit entries with details like author, date, and description. The right pane, titled "Information messages", contains a notice about a release party and a reminder to write wellformed commit messages. Below these panes is a section titled "Employees in this sprint" which displays a grid of employee profiles with their names, specialties, and locations.

Figur 36: viser commitmeldinger til venstre, informasjonsmeldinger oppe til høyre og ansattlisten nede til høyre.

8.1.7 Ansattliste

Helt nytt i nåværende versjon av prototypen er en liste over ansatte med bilde, navn, kunnskap og nåværende lokasjon. Ansattlisten er et direkte resultat av forslag til forbedring

fra iterasjon 2. Spesielt store utviklingsteam vil ha nytte av denne listen, da man i mindre team kjenner bedre til medarbeidernes kunnskap og lokasjon. Lokasjonsinformasjonen i listen er trolig mest nyttig, da lokasjonen kan endres ofte, og kan være vanskelig å være oppdatert på. Informasjonen kan eksempelvis hentes fra brukernes kalendersystemer og således alltid være oppdatert.

8.1.8 Animasjon av ferdige brukerhistorier

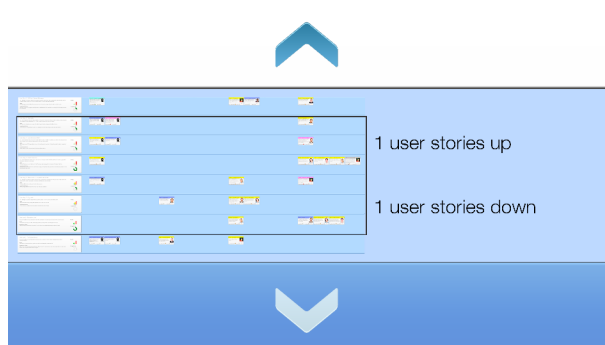
Det ble i denne iterasjonen implementert en funksjon som automatisk flytter ferdige brukerhistorier til bunnen av listen. Funksjonen ble implementert fordi grensesnittet bare viser seks brukerhistorier om gangen, hvilket kan være mindre enn hva det er behov for. Flyttingen er animert slik at det er mulig å se hvor brukerhistorien blir flyttet. Animasjonen starter når det siste oppgavekortet til en brukerhistorie flyttes til ferdigkolonnen. Når brukerhistorien er ferdig animert blir den stemplet «Approved» for ytterligere å signalisere at den er ferdig.

8.1.9 Interaksjon

Interaksjonsmetoden er i denne iterasjonen helt lik den i iterasjon 2, men man trenger ikke lengre å kalibrere FFAST for hver bruker, som beskrevet i seksjon 5.2.2. At kalibreringen er fjernet vil gjøre det mulig å stille seg foran Verdandi og umiddelbart bruke systemet.

8.1.10 Navigasjonsmeny

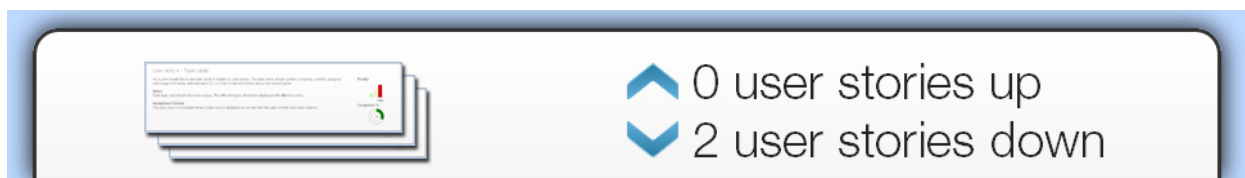
Et viktig fokus i denne iterasjonen var å implementere en god måte å navigere mellom brukerhistorier, da dette fungerte dårlig i forrige iterasjon. Det ble først forsøkt å implementere gester for å navigere mellom brukerhistoriene, men på grunn av begrensninger ved FFAST var ikke dette mulig. Mer spesifikt var denne begrensningen at man ikke kan bevege markøren over flere skjermer og samtidig bruke gester. På grunn av denne svakheten ble det implementert en navigasjonsmeny. Navigasjonsmenyen kan åpnes enten ved å trykke på en navigasjonsindikator eller ved å flytte håndflaten fremover. Navigasjonsmenyen har to knapper, samt informasjon om hvor man har navigert i systemet figur 37.



Figur 37: viser menyen som brukes for å navigere i listen over brukerhistorier. Øverst og nederst i figuren er navigasjonsknappene. Til venstre av senter i figuren illustreres alle oppgavene i systemet, den svarte firkanten viser hvor i listen av brukerhistorier man befinner seg. Til høyre av senter vises hvor mange brukerhistorier som er over eller under nåværende posisjon.

8.1.11 Navigasjonsindikator

I forrige iterasjon kom det klart frem gjennom to funn at det burde være tydeligere informasjon om hvor mange brukerhistorier som var utenfor nåværende skjermbilde. Det ble på bakgrunn av tilbakemeldingene i iterasjon 2 utviklet en navigasjonsindikator på den nederste skjermrekken, illustrert i figur 38.



Figur 38: viser navigasjonsindikatoren, som en knapp som viser hvor mange brukerhistorier som er over og under nåværende utsnitt av informasjonsradiatoren.

8.2 Brukertest

Som i iterasjon 2 ble systemet brukertestet av enkeltindivider med en antatt kunnskap om informasjonsradiatorer sin rolle i smidige metoder. Deltakere ble valgt på bakgrunn av informasjon om at de jobbet med smidige metoder og benyttet informasjonsradiatorer. Brukertesten var lik som i iterasjon 2, men med en del forbedringer. Brukertestens tre hoveddeler og forbedringene presenteres her.

8.2.1 Deltakere

I brukertesten var det totalt fem deltakere, alle menn, fra 25 til 41 år. De ble valgt på bakgrunn av at de hadde kjennskap til smidige metoder og at de benyttet metodikken i sitt daglige arbeid.

8.2.2 Introduksjon og intervju

Brukertesten startet med at det ble lest en introduksjonstekst og deretter ble personalia fylt ut av deltakeren. Etter at personaliaskjemaet var utfyllt fortsatte intervjuet, hvor målet var å avklare deltakernes kjennskap til informasjonsradiatorer, burndowndiagrammer, brukerhistorie- og oppgavekort. Som i forrige iterasjon hadde intervjuet tre hovedspørsmål som hadde oppfølgingsspørsmål for å styre intervjuet i riktig retning.

Oppfølgingsspørsmålene ble i denne iterasjonen endret for å holde intervjuet på riktig sti. Blant annet ble spørsmål som omhandlet prosjekter og opplevelser med informasjonsradiatorer og burndowndiagram fjernet til fordel for flere spørsmål rundt brukerhistorie- og oppgavekort, og koblingen mellom disse.

8.2.3 Opplæring

Opplæringsfasen i denne iterasjonen var identisk med den i iterasjon 2. Deltakeren fikk nødvendige instruksjoner for å kunne navigere i systemet. I tillegg fikk deltakeren prøve blinktest fra første iterasjon for å teste hvordan interaksjonsmetoden fungerte.

8.2.4 Test

Testen startet med at det ble lest en introduksjonstekst til deltakeren, hvor det ble vektlagt hvordan testen ville foregå, forklaring av think-aloud, samt viktigheten av å fortelle alt deltakeren tenkte. Som i iterasjon 2, ble rammene ved brukertesten definert i tråd med retningslinjen til Boren & Ramey (2000).

Sesjonen var guidet rundt tre hovedspørsmål, med oppfølgingsspørsmål og oppgaver. Spørsmålene var utformet slik at de ulike elementene som har blitt forbedret og videreutviklet fra iterasjon 2 ble belyst. Mer spesifikt var de tre hovedspørsmålene:

- Hva er førsteinntrykket av systemet slik som du ser det nå? Hvordan tolker du innholdet?
- Det er generell prosjektinformasjon på skjermene til høyre. Hvordan er framgangen i prosjektet?
- Hvor mange brukerhistorier er det?

Som det fremgår av spørsmålene vil de ikke produsere konkrete svar, men tanker rundt fremgangsmåten for å svare på spørsmålet. Til en hver tid er seks brukerhistorier synlig i Verdandi. Eksempelvis kan det siste spørsmålet besvares med hvor mange brukerhistorier som er synlige eller hvor mange det er i systemet totalt. I tillegg kan spørsmålet besvares ved at man ser på sprintinformasjonen, siden denne viser hvor mange brukerhistorier som gjenstår av en totalsum. Spørsmålene ble stilt på denne måten er for å få deltakerne til å tenke høyt rundt spørsmålet, slik at de kan oppdage muligheten til å skrolle ned og få en bedre forståelse av systemet. Oppfølgingsspørsmål ble stilt for å få en detaljert innsikt i hvordan deltakerne kom fram til de ulike svarene, og for å bekrefte om de forstod elementene slik de var ment.

Oppfølgingsspørsmålene var svært spesifikke slik at ønskede elementer ble dekket hvis disse ikke var godt nok belyst i svarene på hovedspørsmålene. Eksempelvisert var noen av oppfølgingsspørsmålene:

- Hva er de ulike grafene i burndowndiagrammet?
- Hvor mange poeng gjenstår?
- Hvor mange dager er det igjen i denne sprinten?

Alle testsesjonene ble transkribert og senere kategorisert i henhold til et kategoriskjema som er vedlagt i vedlegg C.

8.3 Koding

Råmaterialet fra brukertestene var fem videoer, hver med en varighet mellom 30-45 minutter. Intervju- og testdelen ble transkribert og kategorisert på samme måte som i iterasjon 2.

Som i forrige iterasjon ble hvert intervju oppsummert. Det ble opprettet et kategoriskjema

med utgangspunkt i skjemaet fra iterasjon 2. Kategoriene i skjemaet fra iterasjon 2 ble ansett for å være relevante, men tre nye underkategorier ble lagt til for å dekke nytt innhold og ny funksjonalitet. De nye kategoriene ble identifisert, som i iterasjon 2, ved å individuelt gjennomgå en delmengde av datamaterialet. Et fullstendig kategoriskjema finnes i vedlegg C. Prosessen med å kartlegge brukbarhetsproblemer var identisk med prosessen i forrige iterasjon.

Resultatet av kodeprosessen var 32 uttalelser som var mulige brukbarhetsproblem. Av alle uttalelsene gjaldt 13 innholdet, 13 utseende og 6 interaksjon med systemet. Utenom uttalelsene ble det identifisert 12 forslag til utvidelser eller forbedringer av systemet.

8.4 Analyse

Som i iterasjon 2 vil analysen først vurdere deltakerne i forhold til kunnskapen som ble avdekket i intervjuet. Deretter vil brukbarhetsproblem som ble identifisert vurderes, denne delen av analysen struktureres ut fra de ulike delene i systemet. Brukbarhetsproblemene er utarbeidet ut fra ett eller flere funn, da mange av brukbarhetsproblemene gjelder de samme elementene.

8.4.1 Deltakernes kunnskap

Som en del av intervjuet ble deltakerne bedt om å beskrive hvordan informasjonsradiatoren deres var strukturert og hvordan de brukte den. Kunnskap om informasjonsradiatorer, brukerhistorie- og oppgavekort, samt burndowndiagram ble kartlagt. I denne seksjonen presenteres analysen av intervjuene.

Fire av fem deltakerne hadde brukt en informasjonsradiatorer tidligere. Alle deltakerne var ansatt i ulike IT-selskap. Erfaringsnivået til deltakerne i antall prosjekter hvor en informasjonsradiator var benyttet varierte fra to til over 50. Størrelsen på disse prosjektene var svært variabel.

Deltakerne hadde i all hovedsak brukt informasjonsradiatorer sammen med oppgavekort. Noen deltakere kommenterte likevel at innholdet på informasjonsradiatoren fint kunne være brukerhistorier eller brukstilfeller³⁹. Kun én deltaker hadde brukt en informasjonsradiator med både brukerhistorier og oppgavekort.

³⁹Eng.: use cases

Informasjonsradiatorene som deltakerne hadde brukt var alle delt inn i kolonner, men da med noen ulikheter. Alle informasjonsradiatorene inneholdt kolonnene «skal gjøres», «under utvikling» og «ferdig», noen hadde også «testing». Én deltaker benyttet en informasjonsradiator som var delt opp både i kolonner og rader. Radene representerte da ulike prosjekter, ikke ulike brukerhistorier som i Verdandi. Alle deltakerne hadde kjennskap til burndowndiagram, men én av deltakerne kjente ikke til begrepet velositet. Når begrepet ble forklart viste det seg at han kjente til konseptet velositet, men brukte ikke det konkrete begrepet.

Kun én av fem deltakere hadde brukt Microsoft Kinect, men alle hadde hørt om teknologien tidligere.

8.4.2 Informasjon, burndowndiagram og meldinger

For å fokusere på den daglige interaksjonen viser utsnittet til Verdandi en sprint. Én av deltakerne synes fokuset var for snevert og etterlyste en oversikt av hele prosjektet. Deltakerne hadde ellers ikke problemer med å forstå hvilket nivå de befant seg på. Det var kun en deltaker som ikke klarte å svare på hvilken iterasjon systemet befant seg i, men det tok kort tid før han korrigerste seg og svarte riktig. Ingen av deltakere hadde problemer med å svare på spørsmål om antall brukerhistoriepoeng eller dager som gjenstår.

Det ble kun utført små justeringer på burndowndiagrammet i denne iterasjonen. Selv om det ble brukt andre begreper enn det deltakerne var kjent med, hadde ingen av deltakerne problemer med å tolke informasjonen.

To deltakere påpekte en åpenbar feil ved systemet. Feilen var at meldingene fra kildehåndteringssystemet ble vist i feil rekkefølge, dvs. at de eldste meldingene sto øverst. Ellers var den generelle holdningen at meldinger var et interessant element å ha med, men at det ikke er særlig nødvendig. Ansattlisten var et nytt element, og det var kun én deltaker som hadde vansker med å forstå listen, da han hadde problemer med å finne kompetansen til de ansatte. Resten av deltakerne fant denne informasjonen umiddelbart, og beskrev dette som en nyttig egenskap ved systemet. Én deltaker kom også med forslag om å utvide informasjonen om de ansatte til å inkludere forretningskunnskap.

8.4.3 Brukerhistorier og oppgaver

Én av deltakerne i brukertesten uttalte at han ikke ville brukt kolonnen som representerer oppgaver under testing. Deltakeren mente at dette ikke var nødvendig på en informasjons-radiator, i alle fall ikke for hans utviklingsteam.

Brukerhistorier som er ferdigstilte flyttes automatisk til bunnen av listen ved en animasjon. Tre deltakere stilte seg kritisk til denne funksjonen. Det ble etterlyst en mulighet for å selv kunne bestemme når den ferdige brukerhistorien skal flyttes, da deltakerne ønsket å diskutere brukerhistorien før den ble flyttet. Det var én deltaker som ikke forstod at ferdige brukerhistorier automatisk flyttet seg til bunnen av listen. Deltakeren ble, på grunn av flyttingen, usikker på hvor mange brukerhistorier som befant seg i systemet.

Når en brukerhistorie er flyttet til bunnen av listen, stemples den med «*Approved*». Minst én deltaker syntes stemplingen var misvisende, da ordlyden tilsier at brukerhistorien har blitt godkjent av produkteier eller kunden. Et stempel med teksten «*Done*» ble foreslått av som et bedre alternativ.

Prioriteten til en brukerhistorie presenteres visuelt som et søylediagram med ulike farger, samt en tekstlig beskrivelse. Én av deltakerne ønsket at fargene på søylene hadde høyere kontrast. På spørsmål om hvilke type de ulike oppgavene i systemet representerte, svarte fire av fem deltakere initielt at fargen på oppgavene var relatert til prioriteten. Flere deltakere kommenterte at årsaken til antakelsen var at fargene i søylediagrammet var identisk som bakgrunnsfargen til oppgavekortene.

Koblingen mellom brukerhistorier- og oppgavekort var for svak i iterasjon 2, og var derfor et hovedfokus i denne iterasjonen. Av brukertestene fremgår det at alle deltakerne forstod koblingen, men én deltaker trengte en forklaring for å forstå at elementene var relatert.

På grunn av lengden til MIMIR syntes en deltaker at det var naturlig å stå omtrent to meter unna skjermene. Deltakeren kommenterte derfor at skriften på brukerhistoriene kunne vært større.

8.4.4 Interaksjon

Et brukbarhetsproblem som ble identifisert i første iterasjon var at markøren virret mye på skjermen. Det ble identifisert at virringen skyldtes mappingen av oppløsningen mellom

kamera og MIMIR, og optimalisering ble foretatt i iterasjon 2 for redusere problemet. Likevel var det én deltaker, i denne iterasjonen, som kommenterte at markøren virret. I brukertesten var det en deltaker som kommenterte, ved ett enkelt tilfelle, at markøren på skjermen ikke var synlig da den skjulte seg håndflaten.

Det ble videre ikke identifisert andre brukbarhetsproblemer som går direkte på interaksjonsmetoden, men det ble gitt noen forslag til forbedringer for hvordan interaksjonsmetoden brukes. I denne iterasjonen var pingvinklikk brukt for å velge elementer. Én deltaker ønsket heller å gripe elementer som skulle velges.

Tre deltakere påpekte at navigasjonsmetoden mellom brukerhistorier var for kompleks. Mer spesifikt ble det påpekt at fire handlinger for å navigere ned én rad var for krevende. Det ble foreslått forbedringstiltak hvor man benyttet gester for å utføre den samme navigasjonen.

8.4.5 Utvidelser

Når brukerhistoriene ferdigstilles flyttes disse automatisk til bunnen av listen for å gjøre plass til brukerhistoriene som er under arbeid. En deltaker kommenterte at flyttingen var en god funksjon, men at det kunne være nyttig med en angreknapp. Deltakeren begrunnet angreknappen med at man kanskje flyttet det siste oppgavekortet feilaktig til ferdigkolonnen. Deltakeren foreslo å ha en notifikasjon på øverste skjermrekke, som var synlig i noen sekunder.

Reestimering av oppgaver var svært viktig for én av deltakerne i brukertesten. Deltakeren ønsket å reestimere oppgaver hver gang disse ble flyttet til en ny kolonne. Deltakeren begrunnet dette med at burndowndiagrammet ville være mer oppdatert hvis en oppgave tar mer enn en arbeidsdag.

Samme deltaker foreslo at brukerhistorier og oppgaver burde kunne gjøres større. Videre nevnte deltakeren at det kunne vært interessant å se ulike perspektiver av Verdandi, da man på ulike tidspunkt i løpet av arbeidsdagen var interessert i ulike deler av Verdandi. Deltakeren eksemplifiserte perspektivene ved å si at man på det daglige scrummøtet var interessert i all informasjon som vises på Verdandi, mens man i løpet av arbeidsdagen kun var interessert i å flytte oppgavene.

8.5 Diskusjon

I denne diskusjonen belyses elementene som kom frem fra analysen av brukertestene, og diskusjonen vil derfor følge samme struktur som analysen. Hver seksjon vil presentere mulige årsaker i forhold til de elementer som er beskrevet i analysen, samt funn som er relevante for videreføring av prototypen. Avslutningsvis vil funnene bli oppsummert.

8.5.1 Deltakernes kunnskap

Kunnskapsnivået blant deltakerne var svært høyt, noe som gjenspeiles i intervjuene fra brukertestene. Fire av fem deltakere brukte en informasjonsradiator i sitt daglige arbeid, mens den siste deltakeren brukte en nettbasert versjon. Alle deltakerne kjente til brukerhistorier, oppgavekort og burndowndiagram. Deltakerne gjenspeiler således målgruppen til Verdandi.

8.5.2 Informasjon, burndowndiagram og meldinger

Deltakerne klarte å orientere seg om hvilken dag og hvilket nivå systemet befant seg på, samt svare på hvor mange dager og brukerhistoriepoeng som gjenstod. Verdandi viser tilstrekkelig informasjon om antall dager og brukerhistoriepoeng, men informasjon som viser totalt antall sprinter og prosjektet i sin helhet ble etterlyst. Informasjonsskjermen fungerer således bra, da den samler mye generell informasjon på ett sted. Et annet suksesskriterie er at den er enkelt skrevet i ren tekst, noe som fører til at brukere ikke behøver å tolke ikoner eller annen notasjon for å abstrahere informasjonen. Et funn er derfor at den generelle informasjonen fungerer som en tilleggsfunksjon på informasjonsradiatoren, men en bør også kunne se mer informasjon om prosjektet som helhet. Funnet underbygges av tilbakemelding fra en deltaker som etterlyste å kunne se et større utsnitt av hele prosjektet i tilsvarende format som en sprint vises nå.

Det ble ikke gjort store endringer med burndowndiagrammet fra iterasjon 2, allikevel var forståelsen av diagrammet helt anderledes i denne iterasjonen. At deltakerne hadde god kjennskap til burndowndiagram, gjorde at de ikke hadde problemer med å trekke ut informasjon eller å forstå at diagrammet endret seg underveis. Funn 2 er dermed at et burndowndiagram fungerer bra på digitale informasjonsradiatorer.

Blant deltakerne i denne iterasjonen var det en generell oppfatning at kildekontrollmel-

dingene og ansattlisten er en «grei ting å ha», men at det ikke gir særlig merverdi for systemet som helhet. Oppfattelsen ble begrunnet med at kildekontrollmeldinger er noe man i hovedsak leser for å få innblikk i en spesifikk del av koden, og dermed er enklere å lese på en arbeidsstasjon. Oppfattelsen av ansattlisten var tilsvarende, med begrunnelsen at man som regel kjente til de teknologiske ferdighetene til sine medarbeidere. Likevel ble lokasjonsinformasjonen til de ulike personene i ansattlisten godt mottatt blant deltakerne. En av deltakerne som var positive til ansattlisten ønsket å inkludere forretningskunnskap som en del av informasjonen i listen, da forretningskunnskap var mer aktuelt enn teknologiske ferdigheter. Funn 3 er at synliggjøring av kildekontroll og ansatte ikke er nødvendige elementer på informasjonsradiatoren, men kan være tilleggsfunksjonaliteter.

8.5.3 Brukerhistorier og oppgaver

Én deltaker stilte seg kritisk til å ha med kolonnen *Tasks in testing* på en informasjonsradiator. Ved oppfølgings spørsmål kom det fram at det kunne være nytteverdi ved kolonnen for noen prosjekter, men ikke for prosjektene han var en del av. Uttalelsen identifiserer ikke et direkte brukbarhetsproblem, men gir heller en indikasjon på at man bør kunne velge hvilke kolonner som er synlig på Verdandi. Systemet vil da være mer dynamisk, og vil dermed kunne benyttes av team som jobber på ulik måte. Funn 4 er at det bør være mulighet for å kunne velge hvilke kolonner som er synlig.

Flere deltakere stilte seg kritisk til animasjonen som ble utført når en brukerhistorie ble ferdigstilt. Grunnen til den kritiske innstillingen var at man ofte ønsket å diskutere brukerhistorien før den ble flyttet. Flere deltakere foreslo at man bør kunne selv velge når brukerhistorien skal flyttes, hvilket åpner for diskusjon rundt brukerhistorien og samtidig gir brukeren mer kontroll. Videre ble det påpekt at ordlyden til stempellet på ferdige brukerhistorier var misvisende. Også her kan det være en ide å gi brukeren kontroll over hvordan ferdige brukerhistorier skal stemples. Brukerhistorier kan eksempelvis merkes med «godkjent» hvis kunden har godkjent enheten, eller merkes «til godkjenning» hvis enheten er hos kunden og man venter på godkjenning. Funn 5 er at brukeren selv bør kunne bestemme hvordan ferdige brukerhistorier behandles.

Fire deltakere mente at fargene i søylediagrammet var for lik fargene på oppgavekortene. Intensjonen med like farger var å begrense mengden farger i systemet for å få et ryddig design. Denne intensjonen feilet og fargevalget vurderes som dårlig. Fargevalget bør

utbedres ved videreutvikling av systemet. Selv om nesten alle deltakerne misforstod fargene, klarte samtlige deltakere å identifisere at det var ulike oppgaver i systemet. Fargede oppgavekort, sammen med typeikon og tekstlig beskrivelse gir brukerne av systemet god oversikt over hvilke typer oppgaver som vises på informasjonsradiatoren.

Koblingen mellom brukerhistorier og oppgavekort var ikke tydelig nok i iterasjon 2, og ble derfor utbedret i denne iterasjonen. Brukertestene viste at fire av fem deltakere forstod koblingen mellom brukerhistoriene og oppgavene. Intervjuet og uttalelser fra deltakeren som ikke forstod koblingen, viste at han ikke hadde benyttet en fysisk interaksjonsradiator i de prosjektene han hadde deltatt i. Deltakeren uttalte derimot at koblingen var svært logisk etter at den ble forklart. En av deltakerne som umiddelbart forstod koblingen, benyttet til daglig en informasjonsradiator som hadde tilsvarende inndeling, men hvor hver rad i systemet representerte ulike prosjekter. En annen av deltakerne var svært positiv til inndelingen og uttalte at «Dette er akkurat det vi trenger i mitt team». På bakgrunn av dette er funn 6 at inndelingen mellom brukerhistorier og oppgaver fungerer bra.

8.5.4 Interaksjon

Interaksjonsmetoden ble optimalisert som en del av utviklingen i denne iterasjonen. Optimaliseringen innebar at brukere av systemet ikke trengte å kalibrere kameraet slik som tidligere. Optimaliseringen fungerte særs bra, da deltakerne i brukertesten fikk større bevegelsesfrihet foran Verdandi. Spesielt positivt er optimaliseringen siden systemet som helhet får en mye lavere terskel for bruk, samt gjør aktiviteten ved Verdandi mer naturlig.

Kun to brukbarhetsproblemer med interaksjonsmetoden ble identifisert i denne iterasjonen. Disse problemene omhandlet at markøren virret, og at markøren i ett tilfelle var bakom håndflaten til deltakeren. Begge problemene ble kun identifisert av én deltaker, og sistnevnte problem gjelder kun ved én spesifikk vridning av kroppen hvor skulderleddet og håndflaten er i linje med markøren. Brukbarhetsproblemet er et spesialtilfelle og regnes derfor ikke som alvorlig. At markøren virrer påvirker ikke muligheten til å utføre handlinger i systemet, da de valgbare elementene er av en slik størrelse at interaksjonsmetodens presisjon er mer enn god nok. I lys av dette og resultatene fra iterasjon 2, kan interaksjonsmetoden sis å fungere bra til bruk i Verdandi. Funn 7 er at interaksjonsmetoden med pingvinklikk fungerer sammen med funksjonaliteten til en digital informasjonsradiator.

Likevel kan det være ønskelig å gjøre interaksjonsmetodens handlinger enda mer naturlige. En av deltakerne ønsket å gripe oppgaver fremfor å bruke pingvinklikk. Slike gester er mulig om man står svært nærme kameraet, men ved naturlig avstand fra kameraet vil ikke dybdeinformasjonen som returneres være nøyaktig nok til at dette er mulig. Etter hvert som teknologien utvikles, antas det at dette kan bli mulig i fremtiden. Ved å kunne gripe oppgaver vil interaksjonsmetoden bli nærmere knyttet til den interaksjonen som foregår med en fysisk informasjonsradiator.

Å navigere mellom brukerhistorier ble av deltakerne beskrevet som komplekst. Grunnen til at deltakerne beskrev navigeringen som kompleks er at man må bruke fire bevegelser for å navigere opp eller ned en rad i listen av brukerhistorier. Flere deltakere foreslo å navigere i listen ved hjelp av gester, men på tidspunktet for utviklingen var det ikke støtte for gester i FFAST. Å navigere ved hjelp av gester var foreslått som en initiell løsning på tilsvarende krav i forrige iterasjon, men var ikke mulig med FFAST. Imidlertid har det i ettertid kommet et nytt rammeverk, ZigFu⁴⁰, hvor bruk av gester er mulig. Funn 8 er at navigasjonen må forenkles og effektiviseres for å gjøre brukerhistoriene mer tilgjengelige.

8.5.5 Utvidelser

Det ble gitt fire forslag til utvidelser. Én av deltakerne foreslo å ha en angreknapp, som er en utvidelse som vil være aktuell ved videreutvikling av systemet da det vil gi mulighet til å rette opp feil. I stedet for en knapp kan en gest brukes for å utføre angrehandlingen, hvilket vil styrke det naturlige aspektet ved interaksjonsmetoden.

Å reestimere oppgaver etter hvert steg gir et mer oppdatert burndowndiagram hvis man har oppgaver som strekker seg over mer enn én dag. Forslaget styrker argumentet om at Verdandi bør kunne tilpasses ulike prosjekter. Ingen av de andre deltakerne kommenterte tilsvarende om reestimeringen, noe som kan indikere at ulike prosjekter har ulike behov.

Å ha muligheten til å forstørre elementer, som brukerhistorier og oppgaver, kan være svært fordelaktig. Spesielt gjelder dette når man ønsker å diskutere elementer med andre personer, som under daglig scrum eller ved planleggingsmøter. Ved å kunne forstørre et oppgavekort vil man ha muligheten til å lese dette selv om man står langt unna skjermene og i fellesskap diskutere dette.

⁴⁰ZigFu: et javascript-rammeverk for utvikling av webapplikasjoner som bruker Kinect. Rammeverket har tilsvarende muligheter som FFAST, men muliggjør også bruk av gester i kombinasjon med markørkontroll.

Samme deltaker som foreslo å forstørre elementer, ønsket å se ulike perspektiv på Verdandi i løpet av arbeidsdagen. Deltakeren mente at kun noen av elementene var relevante på ulike tidspunkt. Som presentert i teorikapittelet seksjon 2.1.1 er en intensjon med en informativ arbeidsplass at interessenter hurtig skal kunne få oversikt over prosjektet. Hvis man benytter ulike perspektiv vil denne egenskapen forsvinne ved informasjonsradiatoren. Likevel er det mulig at forslaget ville gjort systemet bedre tilpasset for spesifikke brukere.

8.6 Oppsummering

I denne iterasjonen ble det implementert 24 krav ut i fra resultatet av iterasjon 2. I tillegg til implementasjonen av kravene ble det grafiske grensesnittet forbedret. Forbedringene og utvidelsene av eksisterende elementer ble brukertestet av totalt fem deltakere med god kjennskap til smidige metoder. Brukertestene ble, som i andre iterasjon, filmet og senere transkribert. Transkriptene ble kodet og analysert, som en del av analysen ble deltakerens tolkninger beskrevet i fire kategorier: informasjon, burndowndiagram og meldinger, brukerhistorier og oppgaver, interaksjon og utvidelser. Etter analysen fulgte en diskusjon som ledet til åtte funn:

- Funn 1:** Den generelle informasjonen fungerer som en tilleggsfunksjon på informasjonsradiatoren, men en bør også kunne se mer informasjon om prosjektet som helhet.
- Funn 2:** Burndowndiagram fungerer bra på digitale informasjonsradiatorer.
- Funn 3:** Synliggjøring av kildekontroll og ansatte ikke er nødvendige elementer på informasjonsradiatoren, men kan være tilleggsfunksjonalitet.
- Funn 4:** Det bør være en muligheten for å kunne velge hvilke kolonner som er synlig.
- Funn 5:** Brukeren bør selv kunne bestemme hvordan ferdige brukerhistorier behandles.
- Funn 6:** Inndelingen mellom brukerhistorier og oppgaver fungerer bra.
- Funn 7:** Interaksjonsmetoden med pingvinklikk fungerer sammen med funksjonaliteten til en digital informasjonsradiator.
- Funn 8:** Navigasjonen må forenkles og effektiviseres for å gjøre brukerhistoriene mer tilgjengelig.

Prototypen har de elementer og funksjoner som er beskrevet i teorikapittelet, seksjon 2.1.1, også omtalt som hovedelementene ved en informasjonsradiator. Deltakerne forstod hovedfunksjonaliteten ved Verdandi, men på tross av dette er ikke alle

brukbarhetsproblemer løst. Imidlertid er det svært få av brukbarhetsproblemene som omhandler hovedelementene. Utvidelsesforslagene som ble gitt som en del av brukertestene i denne iterasjonen er elementer som kan påvirke artefaktet som helhet, men som ikke påvirker hovedfunksjonaliteten ved Verdandi og kan være potensielle utvidelser på et senere stadie. Funnene som ble identifisert tas hensyn til ved videre evaluering.

9 Kognitiv dimensjonsanalyse

Som beskrevet i metodekapittelet, seksjon 3.3.2, benyttes det kognitive dimensjonsrammeverket for å vurdere brukbarhet. Rammeverket brukes i dette kapittelet som et diskusjonsverktøy for å evaluere Verdandi.

Sharp et al. (2009) har, som beskrevet i teorikapittelet seksjon 2.1.5, analysert både fysiske kort og informasjonsradiatorer i tråd med de kognitive dimensjonene beskrevet av Green (1989). For å kunne besvare FS1:

Hvordan endres dimensjonene i det kognitive dimensjonsrammeverket når informasjonsradiatoren digitaliseres på MIMIR?,

vurderes prototypen fra iterasjon 3 i forhold til de ulike kognitive dimensjonene, slik at dimensjonene ved Verdandi kan ses i forhold til fysiske informasjonsradiatorer. Videre følger en diskusjon som tar for seg dimensjonenes relasjoner, mediets og interaksjonsmetodens påvirkning, egenskaper ved digitaliseringen, konfigurasjon og tilpasning og synliggjøring av applikasjonen som utvikles. Som et resultat av analysen vil det presenteres retningslinjer for hvordan en digital informasjonsradiator kan utvikles. Avslutningsvis vil retningslinjene og kapittelet oppsummeres.

9.1 Dimensjonene i forhold til Verdandi

Det er, som beskrevet i seksjon 3.3.2, ikke mulig at alle kognitive dimensjoner oppfylles i like stor grad. Analysen av dimensjonene må derfor ta hensyn til hvordan Verdandi skal brukes, hvilket medfører et naturlig kompromiss mellom dimensjonene. Som beskrevet i seksjon 3.3.2, presenterer Blackwell (2000b) fem kategorier for bruk av systemer: *søking*, *legge til*, *modifikasjon*, *avskrivning* og *utforskende*. Verdandi plasseres i kategorien *søking*, men med begrenset interaksjon. Verdandi plasseres i denne kategorien fordi det er et system som skal tilgjengeliggjøre informasjon om prosessen.

Fremgangsmåten for analysen følger de fire stegene som er beskrevet i seksjon 4.6. I forhold til det første steget er Verdandi som helhet hovednotasjonen som analyseres. Verdandi beskrives i seksjon 8.1. Mediet er MIMIR og beskrives i seksjon 5.1. Miljøet hvor

informasjonsradiatoren manipuleres er som beskrevet i seksjon 4.2. Videre finnes det ingen subenheter i prototypen, noe som gjør at det andre steget ikke er gjeldene. Hovedfokuset i analysen er på det tredje steget, som er beskrivelsen av systemet i forhold til de kognitive dimensjonene. Beskrivelsen utføres som en del av denne seksjonen, men er også vedlagt i oppsummert form i vedlegg G. I vedlegget foreligger det også en sammenligning mellom Verdandi og fysiske informasjonsradiatorer. I forhold til det fjerde steget er designmanøvre, som balanserer forholdet mellom dimensjonene, presentert som en del av diskusjonen i dette kapittelet.

Beskrivelse av hver kognitiv dimensjon ble presentert i seksjon 3.3.2. Hvordan fysiske brukerhistorier og informasjonsradiatorer forholder seg til dimensjonene er beskrevet av Sharp et al. (2009) og oversatt i seksjon 2.1.5. De samme 14 kognitive dimensjonene som ble benyttet av Sharp et al. (2009) er vurdert ved Verdandi. Fremgangsmåten for vurderingen er gjennomført som beskrevet i seksjon 4.6. Hver av de 14 kognitive dimensjonene presenteres i det følgende:

Viskositet: Annoteringer i Verdandi har høy viskositet da det ikke har blitt utviklet en mulighet for brukeren til å endre annoteringene i ettertid. Eksempler på slike endringer er å forandre antall kolonner, skifte farger, ta vekk eller legge til elementer på informasjonsradiatoren og kortene.

Det er ikke mulig å flytte flere oppgaver om gangen, men det er derimot tillatt å flytte en oppgave over flere steg. For eksempel ved å flytte et oppgavekort fra *tasks to do* til *tasks done*, uten å måtte være innom *tasks in development* og *tasks in testing*. Når oppgavekortene blir flyttet sorteres de automatisk, og tilsvarende gjelder for brukerhistoriekort som sorteres etter prioritet.

Burndowndiagrammet oppdaterer seg når man flytter et oppgavekort til *tasks done*. Oppdateringen fører til at man slipper å gjøre to handlinger for å holde både informasjonsradiatoren og burndowndiagrammet oppdatert. På samme måte vil gjenstående brukerhistoriepoeng og brukerhistorier også bli oppdatert automatisk. Team som bruker tradisjonelle eller smidige prosjekthåndteringsystemer i kombinasjon med fysiske informasjonsradiatorer og brukerhistorier vil nå kun ha ett system å forholde seg til, og dermed slippe å oppdatere flere steder for én handling. Både automatisk sortering, flytting av oppgaver, oppdatering av burndowndiagrammet og sprintinformasjon forhindrer konsekvensviskositet.

Synlighet: Verdandi begrenser synligheten til prosessen ved at kun seks brukerhistorier vises samtidig. Det er ingen mulighet for å endre utsnitt fra en iterasjon til hele prosjektet, noe som begrenser synligheten til progresjonen i prosjektet. At systemet vises på en videovegg, bevarer den iboende synligheten til informasjonsradiatorer. Et viktig poeng her er at aktiviteten ikke forsvinner slik som med annen programvare. Teammedlemmene må fortsatt gå opp til, og fysisk interagere, med informasjonsradiatoren. Foruten denne aktiviteten er det ikke mulig å se hvem som har flyttet hva i etterkant, men ved et digitalt system kan aktiviteten loggføres, og således øke synligheten til prosessen. Å ha et bilde av utvikleren på en oppgave synliggjør hvem som har ansvaret i større grad enn ved kun å vise et navn, da utenforstående enklere kan koble personer til oppgaver. Applikasjonen blir ikke eksplisitt synliggjort, men commitmeldingene kan gi et innblikk i kodeprosessen.

For tidlige forpliktelser: Verdandi gir ingen forpliktelser til hvordan kildekoden må skrives, men gir noen tidlige forpliktelser i forhold til prosessen. For det første er det ikke mulig å reestimere en oppgave etter hvert steg, dermed vil oppgaven ha samme estimat gjennom hele prosessen. For det andre flyttes en brukerhistorie ned og settes som ferdig når alle oppgavene er flyttet til *tasks done*. At brukerhistorien flyttes ned er et eksempel på for tidlig forpliktelse da det ikke er sikkert at brukerhistorien faktisk er ferdig.

Skjulte avhengigheter: En funksjonell avhengighet mellom brukerhistorier og oppgaver blir uttrykt ved at oppgavene tilhører brukerhistorien på samme rad. Ved å tilhøre en kolonne blir også oppgavene koblet til prosessen. Brukerhistoriekortene viser, med unntak av ferdigstillellesprosenten, ingen avhengighet til prosessen annet enn at de er en del av iterasjonen. Avhengigheter mellom oppgaver, som f.eks. at en oppgave ikke kan løses før en annen oppgave er løst, er skjult. Det er også skjulte avhengigheter mellom commitmeldinger og oppgavekort, da arbeid som utføres bør være koblet til en oppgave. Streng sosial disiplin når man skriver commitmeldinger kan være med på å gjøre denne avhengigheten mer synlig, ved at det skrives eksplisitt hvilke oppgaver som har blitt jobbet med. Siden Verdandi integrerer funksjonaliteten fra fysiske artefakt med programvare, vil det ikke lenger eksistere skjulte avhengigheter hvor man må vedlikeholde to eller flere systemer.

Uttrykksfulle roller: Både representasjons- og innholdsmessig er det en signifikant forskjell på brukerhistorier og oppgaver. Oppgavekortenes plassering i forhold til hvilken rad (brukerhistorie) og kolonne (steg i prosessen) gjør at oppgaven kan tolkes i en større sammenheng, og dermed gi kortet en mer uttrykksfull rolle. De ulike oppgavetyperne

skilles fra hverandre ved bruk av farger, ikon og tekstlig beskrivelse. Utseendemessig er det en liten forskjell på commit- og informasjonsmeldinger, men de skilles fra hverandre i lister som eksplisitt uttrykker rollene til meldingene. Videre vil innholdet i meldingene styrke rollene, da commitmeldinger typisk er av teknisk art, mens informasjonsmeldingene er skrevet med mer naturlig språk.

Utsatt for feil: Verdandi har ingen sikkerhetsmekanismer mot feil i det tekstlige innholdet. Derimot har systemet en begrenset mengde annotasjoner i form av fastsatt layout, bilder, farger og ikoner, og er på grunn av denne begrensningen lite utsatt for feil. Oppgavekort kan bli plassert i feil kolonne, men det er ikke mulig å flytte en oppgave til feil brukerhistorie. Kortene kan ikke mistes, falle ned, eller bli ødelagt, men det kan oppstå problemer med program- eller maskinvare. For å forebygge at data mistes kan man ta sikkerhetskopii, automatisk eller manuelt, av innholdet på informasjonsradiatoren og dermed forebygge feil.

Abstraksjon: Digitale representasjoner av brukerhistorier er abstraksjoner av brukerkrav med ulike abstraksjonsnivå. I applikasjonstermer har brukerhistorier høy abstraksjon, mens oppgaver har lavere abstraksjon da de er mer konkrete i forhold til applikasjonen som utvikles.

I prosestermer har brukerhistorie- og oppgavekort et lavt abstraksjonsnivå. Spesifikt er utvikler sitt bilde, oppgavekortenes farge og ikon eksempler på lavt abstraksjonsnivå på oppgavekortene i Verdandi. Brukerhistorier er prioritert med et tall mellom 0 og 100, men presenteres i et søylediagram som lavt, medium eller høyt prioritert. Denne presentasjonen av prioritering er en abstraksjon av tallene som brukes for å prioritere brukerhistorier. For å endre form og utseende ved Verdandi må man endre kildekoden, det er dermed ingen abstraksjon i form av en konfigurasjon av systemet.

Sekundærnotasjon: Verdandi har en forhåndsbestemt notasjon hvor det ikke er mulig for brukere å legge til sekundærnotasjoner. Primærnotasjonen kan endres, men dette må endres i kildekoden. Det er ikke mulig å legge til sekundærnotasjonelle elementer som merkelapper, men man kan annotere med tekst i brukerhistorier, oppgaver, commit- og informasjonsmeldinger. Tekstannoteringen kan i denne sammenheng være bruk av bokstaver, tall eller tegn, eksempelvis for å referere til andre elementer. Når man skriver commitmeldinger, kan man eksempelvis skrive «Task 1» i begynnelsen av en melding for å gi leserne en indikasjon på at commitmeldingen har en forbindelse til oppgave nummer 1.

Relasjonsnærhet: Kortene er nært relatert til applikasjonen som utvikles da disse representerer delementer i problemdomenet. Kortene er skrevet med naturlig språk av kunden, hvilket ytterligere styrker relasjonsnærheten. Relasjonsnærheten mellom informasjonsradiatoren og prosessen viser seg tydelig, da informasjonsradiatoren som artefakt er laget for å vise progresjon. Digitale artefakt kan øke prosessnærheten, da det er mulig å vise andre elementer som meldinger fra kildehåndteringsverktøy og testrammeverk.

Konsistent: Strukturen, annoteringene, layouten og begrepene har en forutbestemt form, vil være konsistent både innad og mellom ulike team, og kan ikke endres av brukeren. Det er derimot ikke konsistent bruk av farger i systemet, da de samme fargene brukes for å beskrive både prioritet og oppgavens type.

Diffusitet: Notasjonen som brukes i brukerhistoriekort, oppgavekort og burndown-diagram er konsis da den representerer mye informasjon på et lite område av videoveggen. Noen av elementene tar noe stor plass, både for å kunne interageres med og for å kunne leses fra en viss avstand. Elementene er derav noe omstendelige/diffuse. Burndown-diagrammet er et eksempel på et artefakt som gir særs mye informasjon på et lite område, og er dermed konsist.

Vanskelige mentale operasjoner: I forhold til prosessen krever hverken informasjonsradiatoren eller elementer vanskelige mentale operasjoner, men om innholdet i en brukerhistorie er vanskelig å forstå, må meningen ved denne tydes gjennom samtale og diskusjon. Videre reduseres mengden vanskelige mentale operasjoner siden det er mulig å tyde meningen ved brukerhistorier ved å lese oppgavene som er knyttet til denne. Oppgavekort som tilhører en brukerhistorie er plassert på samme rad i systemet, det kreves derfor ingen vanskelige mentale operasjoner for å identifisere hvilke oppgaver som hører til hvilke brukerhistorier.

Burndowndiagrammet gjør det enklere å vurdere om iterasjonen blir ferdig i tide, således reduserer diagrammet mentale operasjoner. Commitmeldingene i Verdandi er symbolannotert med oppgavenummer, hvilket krever at brukeren må lete etter oppgavekortet hvis disse to skal ses i sammenheng. Det kan være vanskelig å tolke applikasjonen ut i fra innholdet på informasjonsradiatoren, og dermed øker kravet til mentale operasjoner.

Midlertidighet: Det er mulig å gjøre det Blackwell & Green (2003) omtaler som «hva om» handlinger, som å flytte en oppgave til *tasks done* for å se hvordan dette påvirker burndowndiagrammet. Midlertidige handlinger er likevel begrenset, da man ikke har muligheten til å ta inn nye brukerhistorier for å se hvordan dette vil påvirke iterasjonen. Strukturen og layouten kan ikke endres og gir klare begrensninger i forhold til kolonner og kortenes størrelse. Kortene har mistet noe av sin fleksible natur og virker mer ferdige, dermed uttrykker kortene mindre grad av midlertidighet enn fysiske kort. Det er, som nevnt under punktet sekundærnotasjon, liten mulighet for å legge til midlertidige annoteringer.

Progressiv evaluering: Fremgangen i iterasjonen vises ved informasjonsradiatorens kolonner, da kolonnene alltid er synlige og viser hvor langt oppgavene er kommet i prosessen. Den generelle sprintinformasjonen viser hvor mange brukerhistorier og brukerhistoriepoeng som er ferdige, og støtter således progressiv evaluering. Progressiv evaluering støttes ytterligere av burndowndiagrammet, da det kontinuerlig holdes oppdatert. Progressiv evaluering styrkes ytterligere ved at brukerhistorier viser en ferdigstillellesprosent basert på hvor mange oppgavekort som er ferdige.

9.2 Diskusjon

Denne diskusjonen tar utgangspunkt i vurderingen av de kognitive dimensjonene, samt Sharp sin analyse for å forsøke å finne sammenhenger, forklaringer, likheter og ulikheter mellom analysene. I tillegg vil analysen knyttes mot teori og tidlige brukertester. En oppsummering av forskjellene mellom den fysiske informasjonsradiatoren og Verdandi er vedlagt i vedlegg G.

I diskusjonen vil dimensjonenes relasjoner og designmanøvre diskuteres. Deretter vil diskusjonen kategoriseres etter hvordan mediet, interaksjonsmetoden og digitaliseringen påvirker dimensjonene. Videre vil det bli diskuteres hvordan mangelen på konfigurasjon påvirker dimensjonene, samt hvordan applikasjonsperspektivet endres. For hver av kategoriene vil det ut i fra funnene presenteres generaliserbare retningslinjer for videre utvikling av informasjonsradiatorer.

9.2.1 Dimensjonenes relasjoner og designmanøvre

Det er, som presentert i seksjon 3.3.2, i følge Blackwell & Green (2003) ikke mulig å se på dimensjonene som isolerte enheter, fordi dimensjonene vil påvirke hverandre. Alle dimensjonene kan ikke være perfekte, men en balanse mellom dimensjonene kan oppnås ved å identifisere designmanøvre (Green & Blackwell, 1998). For å identifisere designmanøvre må relasjonene mellom dimensjonene identifiseres.

Den fysiske informasjonsradiatoren er i følge Sharp et al. (2009) svært *utsatt for feil*, noe som kan forhindres ved digitalisering. Ved å forhindre feilbarheten ved informasjonsradiatoren, blir derimot *midlertidighet* og muligheten til å legge til *sekundærnotasjon* svekket. Konsekvensene av dette er at den digitale informasjonsradiatoren er mindre fleksibel, hvilket er en begrensning av artefaktet. På grunn av begrensningen av artefaktet, bør det fokuseres på å forbedre *midlertidighet* og *sekundærnotasjon*, selv om dette fører til at informasjonsradiatoren kan bli mer *utsatt for feil*.

På samme måte vil det å styrke *abstraksjonsdimensjonen*, i form av bedre konfigurasjonsmuligheter, styrke *viskositeten* ved Verdandi. Ved å åpne for konfigurasjon av Verdandi vil ulike team kunne få et system som er mer tilpasset deres behov, men samtidig vil *konsistensdimensjonen* svekkes. Svekkelsen kan føre til forskjellige grensesnitt mellom ulike team, hvilket kan medføre at systemet/notasjonen blir vanskeligere å lære.

Det finnes flere eksempler på relasjoner mellom dimensjonene og designmanøvre, disse vil bli presentert i de resterende seksjonene.

9.2.2 Mediets og interaksjonenmetodens påvirkning

Verdandi som medie gir naturlig støtte til dimensjoner som *synlighet*, da størrelsen til mediet ivaretar den fysiske representasjonen av en fysisk informasjonsradiator. Både Sharp et al. (2009), Perry (2008), Dubakov & Stevens (2008) og Cockburn (2002) trekker frem *synlighet* som en styrke ved fysiske informasjonsradiatorer. Selv om mediet ivaretar den fysiske representasjonen skaper det utfordringer i forhold til design og utforming. Først og fremst gjelder utfordringene i forhold til rammene på skjermbaserte videovegger. Rammene gjør at innholdet, spesielt tekstlig innhold, ikke bør deles over flere skjermer. I tillegg er det identifisert av Ball & North (2005) at brukere har en tendens til å se på de ulike skjermene som ulike logiske enheter. Dette medførte at hver kolonne med skjermer

ble designet som en logisk enhet f.eks. *tasks to do* og *tasks in development*. I iterasjon 3, jf. funn 6, ble det identifisert at denne inndelingen fungerer bra.

Som Bi & Balakrishnan (2009) hevder kan perifere applikasjoner, som epostklient og kalender, alltid være åpne når man benytter videovegger og dermed øke produktiviteten ved at man bruker mindre tid på å sjekke slike applikasjoner. Sprintinformasjonen, med burndowndiagram og meldinger, kan i Verdandi regnes som en type perifer informasjon, hvilket medfører at MIMIR fasiliteter for *synlighet*.

I tillegg til kolonneinndelingen, påvirker interaksjonsmetoden størrelsen på oppgavekortene. Interaksjonsmetodens påvirkning er todelt: markørens nøyaktighet begrenser størrelsen på oppgavekortene, hvilket fører til at oppgavekortene må ha en viss størrelse for å være mulig å velge; avstanden til sensoren må være omtrent to meter, noe som gjør at elementer må ha en viss størrelse for å være lesbare. Oppgavenes størrelse påvirker derfor *diffusheten* fordi oppgavene er større enn nødvendig for å formidle innholdet.

Mediets og interaksjonsmetodens påvirkning fører til at *synligheten* til brukerhistoriene i iterasjonen reduseres, da kun seks brukerhistorier er synlige simultant. På grunn av at kun seks brukerhistorier er synlige, må det være funksjonalitet for å navigere mellom dem. Funn 8 i iterasjon 3 var at navigasjonen mellom brukerhistoriene må forenkles og effektiviseres for å gjøre brukerhistoriene mer tilgjengelige, hvilket støtter opp om *synligheten* til brukerhistoriene.

Aktiviteten rundt informasjonsradiatoren skal i følge Sharp et al. (2009) og Kurpicz (2011) være *synlig* slik at andre i rommet er oppmerksom på arbeidet som utføres. Synligheten ved Verdandi støtter aktiviteten på tilsvarende måte som ved en fysisk informasjonsradiator, ved at man må interagere direkte med informasjonsradiatoren. Synligheten styrkes ytterligere da man bruker kroppsbevegelser for å interagere, noe som *synliggjør* interaksjonen. I tillegg ble det identifisert i iterasjon 3, jf. funn 7, at interaksjonsmetoden fungerer til formålet. På grunn av mediets fysiske natur og måten man interagerer på ivaretas rollen som en del av den informative arbeidsplassen som beskrevet av Beck & Andres (2004).

Aktiviteten foran informasjonsradiatoren er synlig for samordnede team, men for distribuerte team er aktiviteten skjult. Hvis distribuerte team benytter en digital informasjonsradiator kan aktiviteten synliggjøres ved å bruke en skyggerepresentasjon av brukeren,

slik som presentert av Shoemaker et al. (2010), i seksjon 2.3. Skyggerepresentasjonen i Shoemaker et al. (2010) ble brukt for å plassere brukerens personlige menyvalg, men kan i sammenheng med distribuerte team brukes for å synliggjøre at det er aktivitet på andre lokasjoner.

Basert på diskusjonen over er det utformet følgende retningslinje:

Retningslinje 1: En digital informasjonsradiator bør ivareta aktiviteten og artefaktenes *synlighet*.

9.2.3 Egenskaper ved en digitalisering

En åpenbar egenskap med digitale system er at de kan automatisere handlinger og prosesser, samt automatisk knytte relasjoner mellom tilhørende data. Et eksempel på en handling er å oppdatere burndowndiagrammet, hvilket skjer automatisk i Verdandi. Det eksisterer en *skjult avhengighet* mellom informasjonsradiatoren og burndowndiagrammet. Når man foretar en endring på den fysiske informasjonsradiatoren må man umiddelbart oppdatere burndowndiagrammet for at det fortsatt skal gjenspeile den faktiske prosessen. Hvis ikke dette blir fulgt med streng nøyaktighet vil informasjonen ikke lenger være korrekt, og artefaktet sin hensikt vil falle bort. Å oppdatere et burndowndiagram for hånd vil være en tidskrevende *konsekvensviskositet*, og kan føre til at det blir utsatt, glemt eller ikke utført. Når prosessen automatiseres vil *konsekvensviskositeten* forsvinne, men det vil fortsatt eksistere en *skjult avhengighet* mellom informasjonsradiatoren og diagrammet. Avhengigheten er ikke alvorlig fordi informasjonen automatisk oppdateres, og vil derfor være mindre *utsatt for feil* enn fysiske informasjonsradiatorer. Det at burndowndiagrammet automatisk oppdateres gjør at man enkelt kan gjøre *midlertidige* handlinger for å se hvordan prosjektet vil gå, f.eks. dersom man klarer å utføre en oppgave i løpet av dagen. I tillegg styrker burndowndiagrammet *progressiv evaluering* ved at man kontinuerlig får en tilbakemelding på hvordan prosessen er i forhold til hvor lang tid som gjenstår. I iterasjon 3 ble burndowndiagrammet godt mottatt av deltakerne i brukertesten, jf. funn 2. Diagrammet fungerte bra og er et eksempel på at informasjon kan vises i forskjellige format på digitale informasjonsradiatorer.

Burndowndiagrammet representerer et annet utsnitt av den samme informasjonen som oppgavekortenes posisjon på informasjonsradiatoren, progresjonen i prosjektet visualiseres

altså på to forskjellige måter. I følge undersøkelsen til Azizyan et al. (2011) påpekte 22% av deltakerne som gav tilleggs kommentarer at rapporteringsfunksjoner burde inngå i digitale smidige verktøy. Rapporteringsfunksjoner er i følge Kurpicz (2011) også en populær årsak for å bruke datastøttede verktøy. Dubakov & Stevens (2008) påpeker at en av fordelene med smidige prosjekthåndteringsverktøy er at de har støtte for sanntidsrapportering. Ved å implementere funksjoner for rapportering, kan informasjonen i informasjonsradiatoren deles med interessenter, noe som vil øke *synligheten* til prosessen utad. Ved digitale system kan funksjonalitet for å generere rapporter lages som en *abstaksjon*, og derav ikke forandre informasjonsradiatorens *viskositet*.

En fysisk informasjonsradiator representerer hvordan prosessen ligger an nå, men viser ikke hvordan prosessen har vært. For å bevare hvordan prosessen har vært på fysiske informasjonsradiatorer, kan det tas periodiske utsnitt, som eksempelvis et fotografi av informasjonsradiatoren ved endt arbeidsdag. Som Perry (2008) påpeker er mangelfull loggføring en svakhet med fysiske informasjonsradiatorer. På en digital informasjonsradiator forenkles denne prosessen, da man kan loggføre all aktivitet. Loggføringen kan brukes til å identifisere trender og mønstre i hvordan teamet arbeider (Perry, 2008). Et digitalt system kan dermed øke *synligheten* til den totale prosessen. Et forsøk på å øke *synligheten* til prosessen ble gjort med sprintinformasjonen i Verdandi. Funn 1 var at informasjonen fungerte som en god tilleggsfunksjon på informasjonsradiatoren, men at den også bør kunne vise mer informasjon om prosjektet som helhet. Funn 1 støttes også av to utvidelsesforslag som kom fram i iterasjon 2. Utvidelsesforslagene var at «En bruker skal kunne se oversikt over prosjektet, release og pågående iterasjon» og «En måte å se hele prosjektinformasjonen», begge omhandlet en mulighet for å se ulike nivå av prosessen.

Retningslinje 2: Digitale informasjonsradiatorer bør synliggjøre informasjon i flere forskjellige format, ulike tidsrom og nivå av prosessen.

9.2.4 Konfigurasjon og tilpassing

Som det fremgår i teorikapittelet, seksjon 2.1.1, foreligger det ingen fasitsvar på hvordan en informasjonsradiator skal være utformet med tanke på layout, struktur og innhold. Sharp et al. (2009) sin analyse viser til informasjonsradiatoren som svært fleksibel, der man i stor grad har muligheten til å utføre *midlertidige handlinger* og når som helst kan

legge til *sekundærnotasjon*. Også Dubakov & Stevens (2008) anser fleksibiliteten ved fysiske informasjonsradiatorer som en fordel. Informasjonsradiatorene som ble innhentet fra deltakerne i iterasjon 2 demonstrerer hvordan tre forskjellige grupper studenter med relativt likt utgangspunkt tolker en informasjonsradiator. Også i iterasjon 3 kom det frem at deltakerne hadde ulik praksis ift. layout, merkelapper, inndeling av oppgave- og brukerhistoriekort.

Verdandi, slik den foreligger i dag, har en fast struktur som brukerne ikke har mulighet til å endre. Brukerne har ingen åpenbar mulighet til å legge til *sekundærnotasjon*, unntaket er at man kan skrive direkte på et brukerhistorie- eller oppgavekort. Eksempel på slik *sekundærnotasjon* kan være: «Denne oppgaven kan ikke påbegynnes før oppgave #10 fullføres» eller «Oppgaven vil trolig løse bug1 og bug2 da den oppdaterer databasen». Denne manglende graden av *midlertidighet* og begrensede muligheten for *sekundærnotasjon* vil gjøre systemet mindre fleksibelt enn fysiske informasjonsradiatorer. Ved å ha lik struktur og ikke ha mulighet til å legge til *sækundærnotasjon*, sørger en for at informasjonsradiatoren er *konsistent* mellom ulike team. I iterasjon 3 ble det identifisert et behov for å kunne tilpasse informasjonsradiatoren etter behov, mer spesifikt var funn 4 at brukerne bør kunne velge hvilke kolonner som vises, og funn 5 at man kan bestemme hva som skal skje med ferdige brukerhistorier.

I tillegg til å kunne konfigurere informasjonsradiatoren er det nødvendig å ha funksjoner for å opprette, endre og fjerne brukerhistorie- og oppgavekort. Funksjonene vil styrke Verdandis *midlertidighet* og åpne for en mulighet til å legge til rette for *sekundærnotasjon*. Funksjonaliteten var, som presentert i kapittel 1, utenfor rekkevidden til denne studien. Gjennom studien har det imidlertid kommet forslag til hvordan funksjonene kan implementeres, og presenteres i videre forskning, seksjon 10.2.

Retningslinje 3: Informasjonsradiatoren bør i stor grad være tilpasningsdyktig og kunne konfigureres etter brukerens behov.

9.2.5 Applikasjonen som utvikles

I forhold til applikasjonen som utvikles har brukerhistoriekort høy *abstraksjon*, mens oppgavekort har lav *abstraksjon*. Begge korttypene har *relasjonsnærhet* til applikasjonen som utvikles, men tilsvarende til fysiske artefakt, er det i Verdandi lav *synlighet* i forhold

til applikasjonen som utvikles, fordi applikasjonen uttrykkes av kildekode. Likevel vil Verdandi kunne synliggjøre applikasjonen i større grad enn fysiske informasjonsradiatorer, fordi Verdandi viser commitmeldinger fra kildehåndteringsverktøy. Gitt at meldingene er skrevet på en god måte, vil meldingene kunne gi en representasjon av kildekode, og dermed øke *synligheten* til applikasjonen.

Som presentert i seksjon 2.1.2 hevder Jeffries (2001) at arbeidet som er utført på en applikasjon bekreftes gjennom akseptansetester. Cohn (2004) hevder at det ikke er tilstrekkelig plass på fysiske kort til å skrive akseptansetester. På en digital informasjonsradiator som Verdandi er det rom for å vise tilstrekkelige akseptansetester og koble testene med automatiserte testrammeverk, og således *synliggjøre* applikasjonen.

I følge Azizyan et al. (2011) var integrasjon mot eksisterende systemer det minst tilfredsstillende aspektet ved verktøyene som ble brukt av deltakerne i undersøkelsen, illustrert i figur 4(b). Digitale informasjonsradiatorers mulighet for integrasjon mot kildehåndteringsverktøy og automatiserte testrammeverk, kan i forhold til dette aspektet være fordelaktig. At det er en fordel med denne integrasjonen støttes ytterligere av Dubakov & Stevens (2008), som hevder at integrasjonen er en av fordelene ved smidige prosjekthåndteringsverktøy.

Økt *synlighet* til applikasjonen, vil medføre en økt mulighet for *progressiv evaluering*. Brukerne av digitale informasjonsradiatorer vil ha mulighet til å vurdere applikasjonen i større grad enn på fysiske informasjonsradiatorer, hvor applikasjonsperspektivet er skjult. Samtidig fjernes noen av de *skjulte avhengighetene* mellom prosess og applikasjon ved integrere andre systemer.

Retningslinje 4: Informasjonsradiatoren bør styrke koblingen mot applikasjonen som utvikles.

Som det fremgår av funn 3 var den generelle oppfatningen blant deltakerne at meldinger fra kildehåndteringsverktøyet var «en grei ting å ha med», men at det ikke var nødvendig å ha på en informasjonsradiator. I funn 3 fremgår det ytterligere at meldingene kan inkluderes på en digital informasjonsradiator som valgfri tilleggsinformasjon. I lys av dette bør retningslinjen vurderes i forhold til brukernes behov, i henhold til retningslinje 3.

9.3 Oppsummering

I dette kapitlet ble Verdandi analysert ved det kognitive dimensjonsrammeverket. Resultatet av analysen er en beskrivelse av hvordan notasjonen i Verdandi forholder seg til de 14 dimensjonene i rammeverket.

Det fremgår av resultatet at de kognitive dimensjonene avhenger av hverandre og derfor ikke kan ses som isolerte enheter. Relasjonene mellom dimensjonene ble derfor diskutert i forhold til mediets og interaksjonsmetodens påvirkning, egenskapene ved digitaliseringen, konfigurasjon og tilpassing, samt applikasjonen som utvikles.

Diskusjonen rundt mediets og interaksjonsmetodens påvirkning resulterte i en retningslinje: «En digital informasjonsradiator bør ivareta aktiviteten og artefaktene sin *synlighet*». Bakgrunnen for denne retningslinjen var at Verdandi utvikles på en videovegg som ivaretar rollen som en stor visuell informasjonsradiator. Videre er interaksjonsmetoden svært synlig, hvilket gjør at aktiviteten foran informasjonsradiatoren ikke med enkelhet kan skjules. At rollen og aktivitetene beholdes synlige vurderes som et suksesskriterie ved utvikling av en digital informasjonsradiator.

Videre fremgikk det av diskusjonen et behov for at informasjonen kan vises på flere måter. Argumentet ledet til følgende retningslinje: «Digitale informasjonsradiatorer bør synliggjøre informasjon i flere forskjellige format, ulike tidsrom og nivå av prosessen». I Verdandi blir retningslinjen eksemplifisert ved et burndowndiagram, men kan videreføres i digitale system ved å generere ulike rapporter og visninger av innholdet.

I diskusjonen av konfigurasjon og tilpassing av Verdandi ble det utledet følgende retningslinje: «Informasjonsradiatoren bør i stor grad være tilpasningsdyktig og kunne konfigureres etter brukerens behov». Retningslinjen ble opprettet av to årsaker: fra dimensjonsanalysen ble det identifisert en signifikant forskjell mellom fysiske artefakter og Verdandi i forhold til tilpasning, og fra deltakerne i siste brukertest ble det uttrykt et behov for konfigurering.

Digitale informasjonsradiatorer har potensiale for å synliggjøre applikasjonen som utvikles, noe som ikke enkelt lar seg gjøre med en fysisk informasjonsradiator. Den skjulte koblingen mellom applikasjonen og prosessen utgjør en svakhet ved fysiske informasjonsradiatorer. For å adressere denne svakheten ble det utformet følgende

retningslinje: «Informasjonsradiatoren bør styrke koblingen mot applikasjonens som utvikles». Som beskrevet i diskusjonen må retningslinjen ses i lys av brukernes behov.

Hvordan notasjonsperspektivet ved informasjonsradiatoren endres når den ble digitalisert på MIMIR, besvares gjennom hvordan de kognitive dimensjonene endres. Dimensjonenes endringer ble presentert i dette kapitlet, og en fullstendig oversikt finnes i vedlegg G. Endringene alene besvarer FS1, men sier ikke noe om det kompliserte samspillet mellom dimensjonene. Ved å analysere sammenhengene mellom dimensjonene får man et bedre innblikk i hvordan de kan tas i betraktning ved utvikling av digitale informasjonsradiatorer. Sentrale hovedfunn i denne analysen er hvordan relasjonene mellom de ulike dimensjonene forholder seg til de fire kategoriene i diskusjonen. Hovedfunnene oppsummeres som følgende:

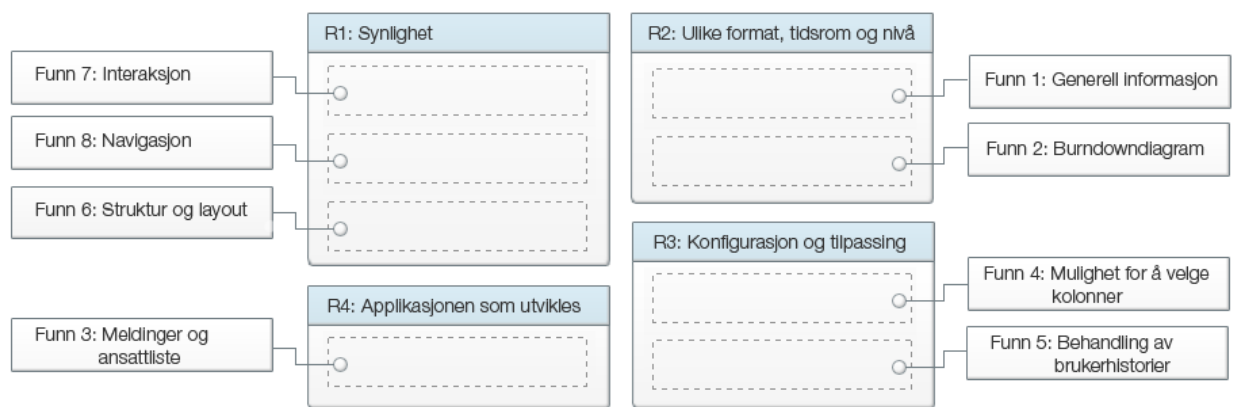
Interaksjonsmetoden og aktiviteten bevarer *synligheten* til informasjonsradiatoren, noe som fører til at Verdandi er tilnærmet *relasjonsnært*, og elementene har samme grad av *diffusitet* og *uttrykksfulle roller*, som fysiske artefakt.

Manglende systemkonfigurering har gjort Verdandi mer *konsistent* og mindre *utsatt for feil*. På grunn av dette økes systemets *viskositet* og krav om *for tidlige forpliktelser*, men samtidig svekkes *midlertidigheten* og muligheten for *sekundærnotasjon* i systemet. Hvis en systemkonfigurering implementeres vil dette være en *abstraksjon* som vil senke systemets *viskositet*, forhindre *for tidlige forpliktelser*, samt fasilitere for *sekundærnotasjon* og *midlertidige handlinger*. Derimot vil en slik konfigurering svekke *synligheten*, *konsistentheten* og skape *skjulte avhengigheter*.

Digitaliseringen har ført til at progresjonen automatisk kan vises på flere måter, hvilket senker *konsekvensviskositeten* fordi den samme informasjonen ikke må oppdateres på flere steder. Denne funksjonaliteten er en *abstraksjon*, som fasiliteter for *progressiv evaluering*, og forhindrer *utsatthet for feil* og *skjulte avhengigheter*.

Fra et applikasjonsperspektiv vil økt *synlighet* til applikasjonen som utvikles fjerne noen *skjulte avhengigheter* og støtte *progressiv evaluering*.

De ulike retningslinjene støttes eller svekkes av funn fra iterasjon 3. Hvilke funn som omhandler de ulike retningslinjene i dette kapitlet illustreres i figur 39.



Figur 39: oppsummerer hvilke funn fra iterasjon 3 som kobles mot de ulike retningslinjene. Retningslinje 1 er forkortet til R1.

10 Konklusjon

I dette kapittelet oppsummeres studien, med dens iterative utvikling og evaluering. Videre presenteres hovedfunnene i studien, og deretter gis en refleksjon av studien og avslutningsvis presenteres forslag til videre forskning.

Hensikten med denne studien var å utvikle og evaluere en digital informasjonsradiator for å få økt innsikt i hvordan man kan kombinere egenskaper fra fysiske informasjonsradiatorer med programvare.

Utgangspunktet og motivasjonen for denne studien var teori som understreket viktigheten av en synlig rolle og aktivitet ved informasjonsradiatorer. For å bevare synligheten ble en MIMIR valgt som medie, hvilket medførte et omfattende arbeid i form av konfigurasjon av MIMIR. Videre ble det identifisert interaksjonsteknologier som var passende til mediet og som samtidig beholdt aktiviteten. For å velge en teknologi og identifisere den best egnede interaksjonsmetoden ble det utført en brukertest som evaluerte interaksjonsmetodene i forhold til effektivitet, presisjon og enkelhet. Resultatet av evalueringen var en interaksjonsmetode som ble brukt videre med prototypen, kalt Verdandi. Prototypen ble utviklet i to iterasjoner, og gjennom utvikling og evaluering ble det identifisert brukbarhetsproblemer, styrker, begrensninger og forslag til utvidelser.

Hovedfunnene i den siste brukertesten var at interaksjon, struktur, statistikk og burndowndiagram ble vurdert som styrker ved systemet. Tre brukbarhetsproblemer ble identifisert: navigasjon mellom brukerhistorier var tungvint, det er et behov for å kunne tilpasse Verdandi og vise ulike nivå av utviklingen. Deltakernes oppfattelse av meldingene fra kildehandlingssystemet var at de var en «grei ting å ha», men ikke vurdert som nødvendig funksjonalitet.

Prototypen ble avslutningsvis evaluert med et kognitivt dimensjonsrammeverk som avdekket hvordan de kognitive dimensjonene endres ved en digitalisering, og på bakgrunn av analysen og resultatene fra brukertesten ble det utarbeidet fire retningslinjer for hvordan man kan utvikle digitale informasjonsradiatorer.

I kapittel 1 ble det presentert to forskningsspørsmål. Formålet til det første forskningsspørsmålet var å belyse hva som endres når informasjonsradiatoren ble digitalisert og

dermed kunne gi bedre innsikt i hvordan informasjonsradiatoren bør digitaliseres:

FS1: *Hvordan endres notasjonsperspektivet ved informasjonsradiatoren når den digitaliseres på MIMIR?*

Forskningsspørsmålet ble besvart som en del av den kognitive dimensjonsanalysen, utført i kapittel 9. Analysen besvarer forskningsspørsmålet ved å identifisere endringene i hver enkel dimensjon, men ga også et innblikk i det komplekse samspillet mellom de kognitive dimensjonene.

Diskusjonen av den kognitive dimensjonsanalysen og funnene i iterasjon 3 førte til følgende retningslinjer for utvikling av digitale interaksjonsradiatorer:

1. En digital informasjonsradiator bør ivareta aktiviteten og artefaktene sin synlighet.
2. Digitale informasjonsradiatorer bør synliggjøre informasjon i flere forskjellige format, ulike tidsrom og nivå av prosessen.
3. Informasjonsradiatoren bør i stor grad være tilpasningsdyktig og kunne konfigureres etter brukerens behov.
4. Informasjonsradiatoren bør styrke koblingen mot applikasjonens som utvikles.

Det andre forskningsspørsmålet i denne studien, skulle identifisere en interaksjonsmetode som ville være kompatibel med MIMIR og samtidig bevare synligheten ved aktiviteten som er foran en fysisk informasjonsradiator. Det ble derfor presentert følgende forskningsspørsmål i kapittel 1:

FS2: *Hvilken interaksjonsmetode er best egnet i kombinasjon med MIMIR for å bevare aktiviteten ved en informasjonsradiator?*

I iterasjon 0 ble det valgt kandidater for interaksjonsteknologier, som ble brukertestet i iterasjon 1. Iterasjonen avdekket at den best egnede interaksjonsmetoden var Kinect med pingvinklikk. Metoden ble brukt og optimalisert i iterasjon 2 og 3, og ble vurdert som egnet til bruk sammen med Verdandi. Flere av interaksjonsmetodene ble forkastet på grunn av mangelfulle tester. Manglende data medfører at forskningsspørsmålet kun *delvis* er besvart og dette adresseres i seksjonene om videre forskning og refleksjoner ved studien.

10.1 Refleksjoner rundt studien

I kvalitative studier kan det være vanskelig å vurdere validitet og reliabilitet (Bryman, 2008), det er derfor gjort flere grep i denne studien for å sikre at datamaterialet holder god kvalitet. Beskrivelse av MIMIR, hvordan utviklingen er utført og en nøyaktig beskrivelse av laboratoriemiljø ble gitt for å kunne utføre studier med like premisser. Verdandi ble brukertestet gjennom tre iterasjoner med totalt 19 deltakere, hvor det i alle iterasjonene ble satt krav om kvalifiserte deltakere.

I iterasjon 2 og 3 ble det brukt semistrukturert intervju som en del av brukertesten. Semistrukturerte intervju skal kunne være repeterbart, derfor bør ikke oppfølgingsspørsmål stilles på en måte som påvirker deltakeren (Sharp et al., 2011). I denne studien anses ikke problemstillingen som særlig relevant, da semistrukturerte intervju først og fremst ble brukt for å kontrollere deltakerens kvalifikasjoner. Deltakernes kvalifikasjoner ble vurdert kontrollert for å sikre datakvaliteten og påliteligheten i observasjonsdelen av brukertesten. Observasjonsdelen ble utført med think-aloudteknikken, som på grunn av sin fleksible natur kan medføre problemer i forhold til validitet og reliabilitet. For å etterstrebe sammenlignbare data, ble det opprettet en testguide. Samtidig ble retningslinjene til Boren & Ramey (2000) fulgt. Brukertestene ble kodet individuelt med samme kategoriskjema uten at det ble identifisert vesentlige forskjeller i resultatet. Dette viser at kodeskjemaet hadde god internreliabilitet. Likevel kan det ikke utelukkes at resultatene ville vært annerledes om andre fagpersoner hadde kodet de samme brukertestene.

I tillegg til brukertester ble det gjennomført en brukbarhetsinspeksjon med et kognitivt dimensjonsrammeverk. At denne brukbarhetsinspeksjonen ble gjort av to personer øker påliteligheten. Likevel kan det, som med kategoriseringen av brukbarhetsproblemer, ikke utelukkes at resultatene ville vært annerledes om andre fagpersoner hadde utført analysen.

At fremgangsmåten for den kognitive dimensjonsanalysen til Sharp et al. (2009) ikke ble beskrevet i deres studie, gjør det uklart hvordan de gikk frem for å evaluere de fysiske artefaktene. For å gjøre evalueringen med det kognitive dimensjonsrammeverket enklere å reproducere, beskrives fremgangsmåten i seksjon 4.6. For å øke kvaliteten til evalueringen kunne det med fordel vært brukt eksterne eksperter, men dette var ikke mulig med tiden og med de ressursene som var til rådighet.

Studien kombinerer brukertester og brukbarhetsinspeksjon. Dette styrker studiens

validitet, da flere kilder benyttes for å besvare forskningsspørsmålene. For å ytterligere styrke validiteten kan Verdandi vurderes i et naturlig miljø, hvor systemet kan bli brukt i daglig arbeid.

10.2 Videre forskning

Evaluering av Verdandi har foregått i et kontrollert miljø som en del det Hevner & Chatterjee (2010) beskriver som designsyklusen. Etter designsyklusen er ferdig skriver Hevner at artefaktet skal føres tilbake til miljøet hvor behovet for artefaktet har sitt opphav. I et naturlig miljø kan det sosiale perspektivet, som beskrevet av Sharp et al. (2009), ved Verdandi evalueres gjennom observasjon. I en slik evaluering kan det vurderes hvordan Verdandi vil fungere i reelle prosjekter og kunne gi et grunnlag for en mer fullstendig sammenligning mellom fysiske og digitale informasjonsradiatorer.

Som nevnt i introduksjonen, kapittel 1, er prototypen begrenset og inkluderer ikke plannleggingsfasen der brukerhistoriene opprettes. Hvis et system som Maurers «AgilePlanner» kombineres med Verdandi, vil hele livssyklusen til brukerhistorier og oppgaver støttes. Et annet alternativ er å benytte nettbrett for å redigere og opprette brukerhistorier og oppgaver. Miller et al. (2008) benyttet, som beskrevet i seksjon 2.2, en PDA for interagere med videovegger. Nettbrett kan i likhet med PDA brukes for å redigere tekstlige elementer. Nettbrettes kamera kan også brukes til å gjenkjenne brukerhistorier og oppgaver, eksempelvis ved å bruke objektgjenkjenning eller mer primitive QR-koder⁴¹. Det foreslås derfor som videre forskning å vurdere hvordan nettbrett kan brukes for å interagere med videovegger, og i den sammenheng inkludere hele livssyklusen til brukerhistorier og oppgaver.

I iterasjon 1 ble ulike interaksjonsmetoder vurdert med et testsett. FS2 ble bare delvis besvart, og for å fullt besvare dette spørsmålet bør det utvikles flere og bedre tester. Videre forskning bør ta for seg flere tester, særlig med fokus på gester og hoverering som presterte bra i iterasjon 1. Spesielt interessant kan det være å vurdere Microsoft Kinect med ulike rammeverk som støtter ulik interaksjon. Rammeverket ZigFu, støtter eksempelvis bruk av gester i større grad enn FFAST v0.7 som ble brukt i denne studien. Videre forskning på gester bør også vurdere annen teknologi, eksempelvis Leap Motion⁴². Leap er en ny

⁴¹QR-kode: forkortelse for «Quick Response-kode» som er en todimensjonal strekkode.

⁴²Leap Motion: en modul som støtter nøyaktig naturlig interaksjon. <http://www.leapmotion.com/>

teknologi som lanseres i løpet av årsskiftet 2012/2013, hvor nøyaktigheten er målbar ned til 0.01 mm. Med en slik nøyaktighet kan avanserte fingergester benyttes for å interagere, samt at nøyaktigheten kan gjøre det mulig å velge svært små elementer, noe som er en svakhet ved Kinect.

Retningslinje 4 var «informasjonsradiatoren bør styrke koblingen mot applikasjonens som utvikles». Å styrke koblingen kan oppnås ved å synliggjøre informasjon fra kildehandtering, testrammeverk og lignende systemer som gir tilbakemelding om applikasjonens tilstand. Å synliggjøre applikasjonens tilstand vil være et steg mot integrasjon av hele livssyklusen til utviklingen og samtidig støtte Dubakov & Stevens (2008) sin andre prediksjon: «Systemer for smidig prosjekthåndtering utvikler seg til løsninger som integrerer hele livssyklusen ved utvikling». En mulig problemstilling ved videre forskning er å kartlegge hvilken informasjon fra andre systemer som kan synliggjøre applikasjonen på en digital informasjonsradiator.

10.3 Avslutningsvis

Beck & Andres (2004) ble innledningsvis sitert da de hevder at digitale brukerhistorier ikke gir en brøkdel av verdien til fysiske brukerhistorier på en fysisk vegg. Ved å ivareta synligheten, fleksibiliteten, gi ulike utsnitt og synliggjøre applikasjonen i tråd med forslagene ovenfor, vil en ikke bare gi brøkdelen av, men *øke* verdien til digitale informasjonsradiatorer.

Referanser

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods. Review and analysis. Teknisk rapport 477, VTT Publications, Espoo.
- Advanced Micro Devices (2010). ATI Radeon HD 5870 Eyefinity 6 Edition Graphics. Hentet fra <http://www.amd.com/us/products/desktop/graphics/ati-radeon-hd-5000/hd-5870-eyefinity-6-edition/pages/overview.aspx>.
- Andrews, C., Endert, A., Yost, B., & North, C. (2011). Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization*, 10(4), 341–355.
- Azizyan, G., Magarian, M. K., & Kajko-Matsson, M. (2011). Survey of Agile Tool Usage and Needs. *AGILE Conference (AGILE), 2011*, (s. 29–38). IEEE.
- Ball, R. & North, C. (2005). Analysis of User Behavior on High-Resolution Tiled Displays. M. Costabile & F. Paterno (Red.), *Human-Computer Interaction - INTERACT 2005*, volum 3585 av *Lecture Notes in Computer Science* (s. 350–363). Springer.
- Ball, R., North, C., & Bowman, D. A. (2007). Move to improve: promoting physical navigation to increase user performance with large displays. *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '07*, (s. 191–200)., New York, NY, USA. ACM.
- Beck, K. & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (andre utg.). Boston, MA, USA: Addison-Wesley.
- Bi, X. & Balakrishnan, R. (2009). Comparing usage of a large high-resolution display to single or dual desktop displays for daily work. *Proceedings of the 27th international conference on Human factors in computing systems, CHI '09*, (s. 1005–1014). ACM.
- Blackwell, A. (2000a). Dealing with new Cognitive Dimensions. *Workshop on Cognitive Dimensions: Strengthening the Cognitive Dimensions Research Community. University of Hertfordshire*.

- Blackwell, A. (2000b). Human Computer Interaction Notes - Advanced Graphics & HCI. Hentet fra: <http://www.cl.cam.ac.uk/teaching/2000/AGraphHCI/HCI/hcinotes.html#cds>.
- Blackwell, A. & Green, T. (2003). Notational systems—the cognitive dimensions of notations framework. J. Carroll (Red.), *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, The Morgan Kaufmann Series in Interactive Technologies (s. 103–134). San Francisco, CA, USA: Morgan Kaufmann.
- Boren, T. & Ramey, J. (2000). Thinking aloud: reconciling theory and practice. *Professional Communication, IEEE Transactions on*, 43(3), 261–278.
- Brown, J., Lindgaard, G., & Biddle, R. (2008). Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts. *Agile, 2008. AGILE '08. Conference*, (s. 39–50).
- Bryman, A. (2008). *Social research methods* (tredje utg.). Oxford: Oxford University Press.
- Cao, X. & Balakrishnan, R. (2003). VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. *Proceedings of the 16th annual ACM symposium on User interface software and technology, UIST '03*, (s. 173–182)., New York, NY, USA. ACM.
- Cockburn, A. (2002). *Agile software development*. Boston, MA, USA: Addison-Wesley.
- Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Boston, MA, USA: Addison-Wesley.
- Desurvire, H. (1994). Faster, cheaper!! Are usability inspection methods as effective as empirical testing? *Usability inspection methods*, (s. 173–202). John Wiley & Sons, Inc.
- Dubakov, M. & Stevens, P. (2008). Agile Tools. The Good, the Bad and the Ugly. Teknisk rapport, TargetProcess, Inc. Hentet fra <http://www.targetprocess.com/download/whitepaper/agiletools.pdf>.
- Dyba, T. & Dingsoyr, T. (2009). What Do We Know about Agile Software Development? *IEEE Software*, 26(5), 6–9.
- Ebert, A., Thelen, S., Olech, P.-S., Meyer, J., & Hagen, H. (2010). Tiled++: An Enhanced Tiled Hi-Res Display Wall. *IEEE Transactions on Visualization and Computer Graphics*, 16(1), 120–132.

- Gallo, L., Placitelli, A. P., & Ciampi, M. (2011). Controller-free exploration of medical image data: Experiencing the kinect. *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, (s. 1–6).
- Gjørøseter, T. & Krzywinski, A. (2012). Mimir: About. Hentet (10.03.2012) fra: <http://www.facebook.com/pages/MIMIR/108270615888289?sk=info>.
- Green, G. C. & Hevner, A. R. (2000). The Successful Diffusion of Innovations: Guidance for Software Development Organizations. *IEEE Software*, 17(6), 96–103.
- Green, T. R. G. (1989). Cognitive dimensions of notations. *Proceedings of the fifth conference of the British Computer Society, Human-Computer Interaction Specialist Group on People and computers V*, (s. 443–460)., New York, NY, USA. Cambridge University Press.
- Green, T. R. G. (1991). Describing information artefacts with cognitive dimensions and structure maps. D. Diaper & N. V. Hammond (Red.), *Proceedings of HCI91: Usability Now, Annual Conference of BCS Human-Computer Interaction Group.*, volum VI (s. 297–316). New York, NY, USA: Cambridge University Press.
- Green, T. R. G. & Blackwell, A. (1998). Cognitive dimensions of information artefacts: a tutorial. Version 1.2. Hentet fra: <http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDtutorial.pdf>.
- Hevner, A. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2), 87–92.
- Hevner, A. & Chatterjee, S. (2010). *Design Research in Information Systems: Theory and Practice*, volum 22 av *Integrated Series in Information Systems*. New York, NY, USA: Springer.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Jeffries, R. (2001). Essential XP: Card, Conversation, Confirmation. Hentet fra: <http://xprogramming.com/articles/expcardconversationconfirmation/>.
- Jeffries, R. & Desurvire, H. (1992). Usability testing vs. heuristic evaluation: was there a contest? *SIGCHI Bulletin*, 24(4), 39–41.
- Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. *Proceedings of the SIGCHI conference on*

- Human factors in computing systems: Reaching through technology*, CHI '91, (s. 119–124)., New York, NY, USA. ACM.
- Karat, C. (1994). A comparison of user interface evaluation methods. J. Nielsen & R. L. Mack (Red.), *Usability inspection methods* (s. 203–233). New York, NY, USA: John Wiley & Sons, Inc.
- Krahmer, E. & Ummelen, N. (2004). Thinking about thinking aloud: A comparison of two verbal protocols for usability testing. *Professional Communication, IEEE Transactions on*, 47(2), 105–117.
- Kurpicz, M. (2011). *Tool Support for Scrum*. Bacheloroppgave i Informatikk, Universitetet i Bern, Bern.
- Lee, J. (2008). Johnny Lee demos Wii Remote hacks. [Video]. Hentet fra http://www.ted.com/talks/johnny_lee_demos_wii_remote_hacks.html.
- Malik, S., Ranjan, A., & Balakrishnan, R. (2005). Interacting with large displays from a distance with vision-tracked multi-finger gestural input. *Proceedings of the 18th annual ACM symposium on User interface software and technology, UIST '05*, (s. 43–52)., New York, NY, USA. ACM.
- Mewhinney, M. & Fay, M. (2008). NASA Develops World's Highest Resolution Visualization System . Hentet fra: http://www.nasa.gov/centers/ames/news/releases/2008/08_57AR.html.
- Miller, C., Robinson, A., Wang, R., Chung, P., & Quek, F. (2008). Interaction techniques for the analysis of complex data on high-resolution displays. *Proceedings of the 10th international conference on Multimodal interfaces, ICMI '08*, (s. 21–28)., New York, NY, USA. ACM.
- Ni, T., Schmidt, G. S., Stadt, O. G., Livingston, M. A., Ball, R., & May, R. (2006). A survey of large high-resolution display technologies, techniques, and applications. *Proceedings of the IEEE conference on Virtual Reality, VR '06*, (s. 223–236)., Washington, DC, USA. IEEE Computer Society.
- Nielsen, J. (1994a). Estimating the number of subjects needed for a thinking aloud test. *International Journal of Human Computer Studies*, 41(3), 385–397.

- Nielsen, J. (1994b). Heuristic evaluation. *Usability inspection methods* (s. 25–62). New York, USA: John Wiley & Sons, Inc.
- Nielsen, J. (1994c). Usability inspection methods. *Conference companion on Human factors in computing systems, CHI '94*, (s. 413–414)., New York, NY, USA. ACM.
- Nielsen, J., Clemmensen, T., & Yssing, C. (2002). Getting access to what goes on in people's heads?: reflections on the think-aloud technique. *Proceedings of the second Nordic conference on Human-computer interaction, NordiCHI '02*, (s. 101–110)., New York, NY, USA. ACM.
- Nielsen, J. & Mack, R. L. (1994). *Usability inspection methods*. New York, NY, USA: John Wiley & Sons Inc.
- Perry, T. (2008). Drifting toward invisibility: The transition to the electronic task board. *Proceedings of the Agile 2008, AGILE '08*, (s. 496–500)., Washington, DC, USA. IEEE Computer Society.
- Robertson, G., Czerwinski, M., Baudisch, P., Meyers, B., Robbins, D., Smith, G., & Tan, D. (2005). The large-display user experience. *IEEE Computer Graphics and Applications*, 25(4), 44–51.
- Sandstrom, T. A., Henze, C., & Levit, C. (2003). The hyperwall. *Proceedings of the conference on Coordinated and Multiple Views In Exploratory Visualization, CMV '03*, (s. 124–133)., Washington, DC, USA. IEEE Computer Society.
- Schuh, P. (2005). *Integrating Agile Development In The Real World*. Programming Series. Rockland, MA, USA: Charles River Media.
- Schwaber, K. (2004). *Agile Project Management With Scrum*. Redmond, WA, USA: Microsoft Press.
- Sharp, H. & Robinson, H. (2008). Collaboration and co-ordination in mature extreme programming teams. *International Journal of Human-Computer Studies*, 66(7), 506–518.
- Sharp, H., Robinson, H., & Petre, M. (2009). The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers*, 21(1-2), 108–116.

- Sharp, H., Robinson, H., Segal, J., & Furniss, D. (2006). The Role of Story Cards and the Wall in XP teams: A Distributed Cognition Perspective. *Proceedings of the conference on AGILE 2006, AGILE '06*, (s. 65–75)., Washington, DC, USA. IEEE Computer Society.
- Sharp, H., Rogers, Y., & Preece, J. (2011). *Interaction design: beyond human-computer interaction* (tredje utg.). Chichester, Storbritannia: John Wiley & Sons, Inc.
- Shoemaker, G., Tsukitani, T., Kitamura, Y., & Booth, K. S. (2010). Body-centric interaction techniques for very large wall displays. *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries, NordiCHI '10*, (s. 463–472)., New York, NY, USA. ACM.
- Suma, E., Lange, B., Rizzo, A., Krum, D., & Bolas, M. (2011). FFAST: The Flexible Action and Articulated Skeleton Toolkit. *Virtual Reality Conference (VR), 2011 IEEE*, (s. 247 –248).
- The Linux Information Project (2006). The X Window System:A Brief Introduction. Hentet (14.10.2011) fra: <http://www.linfo.org/x.html>.
- van den Haak, M., De Jong, M., & Jan Schellens, P. (2003). Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue. *Behaviour & Information Technology*, 22(5), 339–351.
- van Someren, M. W., Barnard, Y. F., & Sandberg, J. (1994). *The think aloud method : a practical guide to modelling cognitive processes*. London, Storbritannia: Academic Press.
- Wake, W. (2002). *Extreme Programming Explored*. Boston, MA, USA: Addison-Wesley.
- Wang, X. & Maurer, F. (2008). Tabletop AgilePlanner: A tabletop-based project planning tool for agile software development teams. *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, (s. 121–128).

Vedlegg A Kravkatalog for iterasjon 2

Kravnr	Krav	Prioritet	Estimat
1	Brukerhistorier skal inneholde kort overskrift, en tekstlig beskrivelse, prioritet, akseptanskriterie og notater	75	1
2	Brukerhistorier skal være sortert etter prioritert rekkefølge	50	4
3	Brukerhistorier skal ha en ferdigstillellesprosent. Prosenten skal vise antall oppgavekort som er ferdige i forhold til hvor mange som er under utvikling eller ikke påbegynt.	55	4
4	Brukerhistorier skal skrives på formen: som en <type bruker>, vil jeg <et mål> slik at <en grunn>	70	1
5	Oppgavekort skal vise estimat, utviklernavn og bilde, oppgave identifikator, samt tekstlig beskrivelse.	65	1
6	Oppgavekort skal være visuelt knyttet til brukerhistorier.	95	8
7	Ha ulike typer oppgavekort, eksempelvis for koding, brukergrensesnitt, og bugs.	60	2
8	Det skal være totalt fire kolonner for å visualisere prosessen, dette inkluderer <i>to do</i> , <i>in progress</i> , <i>testing</i> og <i>done</i>	100	4
9	Det skal være tydelig hvor i prosessen et oppgavekort befinner seg.	90	2
10	Det skal være mulig å flytte et oppgavekort fra en kolonne til en annen, men det skal ikke være mulig å flytte en oppgave mellom brukerhistorier.	80	8
11	Det skal være mulig å flytte oppgavekort frem og tilbake mellom kolonner.	85	4
12	Det skal være en burndowndiagram over nåværende sprint. Denne grafen skal oppdateres automatisk når en oppgave er utført, og skal alltid skal være synlig.	35	8
13	Vise statistikk om hvordan sprinten ligger an: dette innebærer å vise antall gjenværende dager og storypoints.	5	4
14	Vise de siste endringene i prosjektet	10	4
15	Vise generelle meldinger. Eksempelvis hvem som ikke er på jobb og om det skal være et møte de nærmeste dagene	15	1
16	Det skal være mulig å navigere opp og ned i en liste av brukerhistorier. Det skal da gis tilbakemelding om hvor man er i denne listen	45	4
17	Det skal være mulig å se hvor mange brukerhistorier som er ferdige, selvom de ikke vises i nåværende skjermbilde	40	2

Tabell 3: Kravkatalog for iterasjon 2. Tabellen viser kravnummer, krav, prioritet og estimat.

Vedlegg B Kravkatalog til iterasjon 3

Kravnr	Krav	Prioritet	Estimat
1	En bruker skal umiddelbart vite hvilket nivå (prosjekt, release eller sprint) man befinner seg på	Middels	4
2	En bruker skal kunne se hva som er korrekt tidsenhet.	Høy	1
3	Begrepet «Optimal development» må endres.	Middels	1
4	Grensesnittet bør vise hvor mange brukerhistoriepoeng det var opprinnelig.	Høy	1
5	Grensesnittet bør vise hvor mange dager det var opprinnelig og hvor mange dager som gjenstår i nåværende sprint.	Høy	1
6	Regular messages bør endres til en mer forklarende tekst.	Lav	1
7	Innholdet til regular messages bør endres.	Lav	1
8	Grensesnittet bør vise hvor mange brukerhistorier det er i nåværende periode	Høy	2
9	Endre brukerhistoriekortene slik at informasjonen blir mer oversiktlig	Middels	4
10	Tydeliggjøre brukerhistoriers prioritet	Middels	2
11	Brugerhistoriekortenes poengsum skal stemme med totalsummen av oppgavekortenes poeng.	Middels	2
12	Tydeliggjøre koblingen mellom brukerhistorier og oppgaver.	Høy	8
13	Ha lik notasjon mellom oppgavekort	Høy	2
14	Oppgavekortenes tekstlige typebeskrivelse kan synliggjøres mer	Middels	2
15	Grafisk fremstille at en brukerhistorie er ferdig	Middels	4
16	Flytte ferdige brukerhistorier til bunnen av listen	Lav	4
17	Et oppgavekort skal sorteres når den slippes fra en kolonne til en annen.	Høy	8
18	Utvikle en navigasjonsmetode hvor man enkelt kan navigere blant brukerhistoriene	Høy	8
19	Indikator som viser at det er flere brukerhistorier lenger ned/opp	Middels	4
20	Vise grafisk/hindre når man er kommet til slutten av listen med brukerhistorier.	Lav	2
21	Systemet skal vise hvem som jobber på prosjektet	Lav	2
22	Systemet skal vise kompetansen til de ulike utviklerne	Lav	2
23	Commitmeldingene bør generaliseres.	Middels	1
24	Implementere en funksjonalitet for å se hvem som har flyttet oppgavekortene	Lav	8

Tabell 4: Kravkatalog til iterasjon 3. Tabellen viser kravnummer, krav, prioritet og estimat.

Vedlegg C Kategoriskjema til koding av brukbarhetsproblemer

Vedlegget viser kategoriskjemaet som ble brukt i iterasjon 2 og 3. Kategoriskjemaet ble skrevet som en del av iterasjon 2, og utvidet i iterasjon 3. Ingen kategorier ble fjernet i iterasjon 3. Kategorier som ble lagt til i iterasjon 3 er merket * .

1. Innholdsproblemer

1. Uttalelser viser usikkerhet om brukerhistoriekort sin rolle i systemet.
2. Uttalelser viser usikkerhet om oppgavekort sin rolle i systemet.
3. Uttalelser viser usikkerhet hvorvidt veggen viser informasjon om en sprint, release eller prosjektet som helhet.
4. Uttalelser viser at deltakeren har problemer med å tolke informasjonen fra burndowndiagrammet.
5. Uttalelser viser usikkerhet om hvor mange story points og/eller antall dager som gjenstår i prosjektet.
6. Uttalelser viser at deltakeren har problemer med å forstå commit- og/eller generelle meldinger.
7. * Uttalelser viser at deltakeren har problemer med inndelingen av kolonner.
8. * Uttalelser viser at deltakeren har problemer å finne informasjon om de ansatte.
9. * Uttalelser viser at deltakeren er usikker på hva som skjer med en brukerhistorie når den er ferdig.

2. Layoutproblemer

1. Uttalelser viser at deltakeren har problemer med utseendet og/eller innholdet til en brukerhistorie.
2. Uttalelser viser at deltakeren ikke forstår koblingen mellom brukerhistoriekort og oppgavekort.
3. Uttalelser viser usikkerhet rundt oppgavene sin farge/type.
4. Uttalelser viser at deltakeren mistolker hvor mange brukerhistorier som finnes

3. Usikkerhet

1. Uttalelser viser usikkerhet om kunnskap hos deltakeren.
2. Uttalelser viser usikkerhet om deltakeren kan bevege seg fritt.

4. Forslag

1. Deltakeren gir forslag til forbedringer.
2. Deltakeren gir forslag til utvidelser.

5. Interaksjonsproblemer

1. Uttalelser viser usikkerhet om å velge og slippe elementer.
2. Deltakeren har problemer med å navigere mellom brukerhistorier.
3. Deltakeren kommenterer at markøren beveger seg på uønsket vis.
4. Deltakeren klarer ikke på egen hånd å hente seg inn igjen etter det har oppstått en feil.

Vedlegg D Mal for brukertest - Iterasjon 1

Iterasjon 1 - Brukertesting av interaksjonsmetoder

Leses til testpersonen før testene begynner: «Takk for at du har tatt deg tid til å komme her i dag. Vi skal i dag teste to ulike interaksjonsteknologier til systemet som du ser på veggen. Det vi er ute etter å oppdage er hvordan ulike interaksjonsmetoder føles og oppleves, vi vil også måle presisjonen til interaksjonsmetodene når du utfører oppgaver. Du vil få instruksjoner om hva du skal gjøre før hver oppgave.

Oppgavene du utfører i dag vil bli filmet slik at vi senere kan gå tilbake å analysere eventuelle problemer ved systemet. Kameraet er derfor rettet mot systemet og ikke mot deg. Du vil holdes anonym i studien som vi utfører, og ditt navn eller identitet vil ikke bli utgitt.

Etter noen av oppgavene vil vi spørre om hvilken kontrollfølelse du hadde når du utførte testen. Kontrollen måles på en skala fra én til fem, der én er ingen kontroll og fem er full kontroll. Dette vil bli repetert når det er nødvendig. Hvis noe er uklart underveis i testen, er det bare å spørre meg. Hvis du ønsker å avbryte en oppgave eller observasjonen, sier du ifra til meg.»

Detaljer om deltaker

Dato: __. __. __ Navn: _____ Alder: ____ Kjønn: M / K

Yrke: _____ Studie: _____

Har brukt gyromus tidligere: Ja / Nei

Har brukt akselerometermus/mus på iPhone tidligere: Ja / Nei

Har brukt/spilt Microsoft Kinect: Ja / Nei

Gyromus

Leses til deltaker: «Du skal nå få utføre noen oppgaver hvor du skal bruke en gyromus. Dette er en mus som reagerer på g-krefter. Det vil si at musepekeren beveger seg i forhold til hvordan du beveger gyromusen. *Vis deltakeren hvordan man bruker gyromusen.* For å aktivere pekeren må du holde inne knappen på undersiden. Hvis du dobbelklikker på knappen på undersiden vil musepekeren være aktiv selv om du ikke holder den inne» *Vis*

hvordan knappen på undersiden av gyromusen fungerer.

Knappetest

Les til deltaker: «Beveg markøren slik at den er over en av de røde firkantene om gangen. Hold markøren i ro i to sekunder til den blir grønn. Markøren skal ikke forlate firkanten i disse 2 sekundene. Begynn med venstre boks, forsett mot høyre. Ikke ta pauser mellom knappene.»

Resultat knappetest:

Topp venstre (Stor): _____	Bunn venstre (Stor): _____
Topp sentrert (Stor): _____	Bunn sentrert (Stor): _____
Topp høyre (Stor): _____	Bunn høyre (Stor): _____
Topp venstre (Middels): _____	Bunn venstre (Middels): _____
Topp sentrert (Middels): _____	Bunn sentrert (Middels): _____
Topp høyre (Middels): _____	Bunn høyre (Middels): _____
Topp venstre (Liten): _____	Bunn venstre (Liten): _____
Topp sentrert (Liten): _____	Bunn sentrert (Liten): _____
Topp høyre (Liten): _____	Bunn høyre (Liten): _____

Blinktest

Leses til deltaker: «Her skal det skytes på blink. Velg siktet og skyt ved å slippe siktet over målskiven. Det er lov til å slippe siktet underveis, men da må dette skje utenfor målskiven. Det vil være tre runder.»

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Liten blinktest

Leses til deltaker: «Denne testen er lik som den forrige, men blink og sikte er mindre.

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Biltest

Leses til deltaker: «Flytt bilen langs veien uten å kjøre utenfor veibanen. Skulle du kjøre utenfor er det bare å fortsette etter beste evne innenfor veien. Stopp ved enden av veien og

vent på videre instruksjoner».

Resultat:

Antall utforkjøringer: _____ Kontrollfølelse (1-5): _____

Microsoft Kinect med push

Leses til testpersonen: «Oppgavene i denne delen er helt lik de du utførte tidligere, vi gjør oppgaven om igjen for å kunne sammenligne metodene. Kinect kameraet kan kjenne igjen kroppsdeler, deriblant hendene dine. For å kalibrere denne gjenkjenningen må du stå slik *vis psi-pose*. Når du står slik er det viktig at vinkelen mellom under og overarm er 90 grader».

«Du beveger / navigerer markøren ved å flytte din høyre hånd innenfor kinect kameraet sitt synsrområde. For å klikke på et element tar du høyre hånden fremover mot skjermen. *Vis hvordan*»

Knappetest, Les til deltaker: «Beveg markøren slik at den er over en av de røde firkantene om gangen. Hold markøren i ro i to sekunder til den blir grønn. Markøren skal ikke forlate firkanten i disse 2 sekundene. Begynn med venstre boks, forsett mot høyre. Ikke ta pauser mellom knappene.»

Resultat knappetest:

Topp venstre (Stor): _____	Bunn venstre (Stor): _____
Topp sentrert (Stor): _____	Bunn sentrert (Stor): _____
Topp høyre (Stor): _____	Bunn høyre (Stor): _____
Topp venstre (Middels): _____	Bunn venstre (Middels): _____
Topp sentrert (Middels): _____	Bunn sentrert (Middels): _____
Topp høyre (Middels): _____	Bunn høyre (Middels): _____
Topp venstre (Liten): _____	Bunn venstre (Liten): _____
Topp sentrert (Liten): _____	Bunn sentrert (Liten): _____
Topp høyre (Liten): _____	Bunn høyre (Liten): _____

Blinktest

Leses til deltaker: «Her skal det skytes på blink. Velg siktet og skyt ved å slippe siktet over målskiven. Det er lov til å slippe siktet underveis, men da må dette skje utenfor målskiven.

Det vil være tre runder.»

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Liten blinktest

Leses til deltaker: «Denne testen er lik som den forrige, men blink og sikte er mindre.»

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Biltest

Leses til deltaker: «Flytt bilen langs veien uten å kjøre utenfor veibanen. Skulle du kjøre utenfor er det bare å fortsette etter beste evne innenfor veien. Stopp ved enden av veien og vent på videre instruksjoner».

Resultat:

Antall utforkjøringer: _____ Kontrollfølelse (1-5): _____

Microsoft Kinect med hoverering

Hovereringstest

Leses til deltakeren: «I denne oppgaven skal vi teste hvordan det er å bruke Kinect uten å måtte aktivt gjøre en bevegelse for å velge. I oppgaven skal markøren flyttes over siktet, etter 2 sekunder aktiveres flyttemuligheten. Elementet skal da plasseres så sentrert som mulig på målskiven. Siktet slippes automatisk etter 2 sekunder».

Resultat:

Poeng: _____ Kontrollfølelse (1-5): _____

Microsoft Kinect med gester

Gesttest

Leses til deltaker: «I denne testen skal det benyttes armbevegelser, eller det som kalles gester, for å flytte ballen på skjermen inn målet. Målet vises på høyre del av skjermen.

Testen er over når ballen ikke lenger er synlig. Bruk så få armbevegelser som mulig». *Vise bevegelsene som kan benyttes.*

Resultat:

Antall flytt: _____ Kontrollfølelse (1-5): _____

Microsoft Kinect med pingvin

Leses til testpersonen: «Du beveger/navigerer markøren, som tidligere, ved å flytte din høyre hånd innenfor kinect-kameraet sitt synsområde. For å klikke på et element tar du nå venstre hånden utover fra hoften ca. 20 centimeter. *vis bevegelsen.*

Blinktest

Leses til deltaker: «Her skal det skytes på blink. Velg siktet og skyt ved å slippe siktet over målskiven. Det er lov til å slippe siktet underveis, men da må dette skje utenfor målskiven. Det vil være tre runder.»

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Liten blinktest

Leses til deltaker: «Denne testen er lik som den forrige, men blink og sikte er mindre.»

Resultat:

Runde 1: _____ Runde 2: _____ Runde 3: _____ Kontrollfølelse (1-5): _____

Biltest

Leses til deltaker: «Flytt bilen langs veien uten å kjøre utenfor veibanen. Skulle du kjøre utenfor er det bare å fortsette etter beste evne innenfor veien. Stopp ved enden av veien og vent på videre instruksjoner».

Resultat:

Antall utforkjøringer: _____ Kontrollfølelse (1-5): _____

Vedlegg E Mal for brukertest - Iterasjon 2

Iterasjon 2 - Brukertesting av Verdandi

Takk for at du har tatt deg tid til å komme her i dag. Vi skal i dag teste en digital versjon av en informasjonsradiator, også kjent som en «scrumvegg eller oppgavetavle», for å se hvordan denne fungerer i forhold til en fysisk informasjonsradiator.

Vi vil understreke at dette er en skisse og dette er ikke det ferdige systemet. Det som er viktig for oss er å få tilbakemelding på er de ulike komponentene til systemet og de handlingene som kan utføres.

Vi ønsker at du tenker høyt under hele testen. Det vil si at: hvis det er noe du lurer på, ikke forstår eller hvis du har noe på hjerte så si dette høyt. Du vil få spørsmål og oppgaver underveis i denne testen.

Utførelsen av disse oppgavene vil bli filmet slik at vi senere kan gå tilbake for å analysere eventuelle problemer med dette systemet. Din identitet vil holdes anonym i hele studien. Du vil få instruksjoner om hva du skal gjøre før hver oppgave. Hvis noe er uklart eller hvis du ønsker å avbryte testen er det bare å si ifra til meg. Vi stopper kanskje testen underveis.

Testpersondetaljer

Dato: _____ Navn: _____ Statsborgerskap: _____

Alder: _____ Kjønn: _____ INFO331 prosjektgruppe: _____

Yrke / Studie: _____

Intervju

1. Hva er din kjennskap til begrepet informasjonsradiator?
 - (a) Fortell hvordan du har brukt informasjonsradiator/ scrumvegg tidligere?
 - i. Gi en kort beskrivelse av prosjektet.
 - ii. Hvordan så deres informasjonsradiator ut i layout, fargebruk, notasjon?
 - iii. Hvordan var opplevelsen din av denne informasjonsradiatoren?
 - iv. Var det noe du følte var mindre bra eller du følte du manglet?

2. Kjenner du til begrepet brukerhistoriekort/storycards og oppgavekort?
 - (a) Har du brukt en informasjonsradiator sammen med brukerhistoriekort ?
 - (b) Har du brukt en informasjonsradiator sammen med oppgavekort ?
3. Kjenner du til begrepet burndowndiagram? Forklar gjerne hva du forbinder med det begrepet.
 - (a) Har du brukt et slikt diagram tidligere?
 - (b) Vet du hva burndown velocity er? Viste deres burndowndiagram velocity?
 - (c) Hvordan opplevde du bruken?
4. Har du brukt/spilt kinect tidligere?
 - (a) Prøvd en gang, flere ganger, ofte...evt annet

Opplæring med kinect

Leses til testpersonen: Før vi begynner testen av selve systemet skal vi gi litt opplæring i bruk av kinect. I opplæringen er målet at du blir kjent med interaksjonsmetoden.

1. Vis kalibreringsposituren utenfor kinectkameraets synsvinkel.
2. Vis hvordan musen velges, høyre hånd opp
3. Vis hvordan man styrer musen med håndflaten
4. Vis hvordan man velger et element med pingvinklikk, venstre arm ut 20 cm.
5. Vis hvordan man gir fra seg kontroll, venstre arm opp over hode.
6. La brukeren prøve stor blinktest 3 ganger. Prøv å få brukeren til å bevege siktet litt rundt på skjermene slik at interaksjonsmetoden er forstått.

Poengresultat av stor blinktest. Gjør testpersonen oppmerksom på at: *Plasser siktet så sentrert som mulig over blinken før du slipper siktet. En poengsum vil dukke opp på skjermen.*

Resultat: Runde 1:_____ Runde 2:_____ Runde 3:_____

Test av Verdandi Leses til testpersonen: Vi skal nå begynne å teste prototypen som er under utvikling. Vi vil i denne fasen gi deg ulike spørsmål og oppgaver. Vi ber deg tenke høyt mens du utfører disse oppgavene, slik at vi forstår ditt synspunkt. Skjermene viser utviklingen av et gitt system, se for deg at du nettopp er blitt en del av dette utviklingsteamet. Og at du nå skal orientere deg om prosjektets gang.

1. Hva er førsteinntrykket av systemet slik som du ser det nå? Hvordan tolker du innholdet?

1.1 Er det noe du umiddelbart savner ved systemet som informasjonsradiator?

1.2 Hvordan tolker du inndelingen av brukerhistoriekort og oppgavekort?

1.2.1 Hvordan tolker du koblingen mellom brukerhistoriekort og oppgavekort?

1.2.2 Hvordan tolker du de ulike fargene på oppgavene?

2. Det er generell prosjektinformasjon på skjermene til høyre. Hvordan er framgangen i prosjektet?

2.1 Ser det ut til at prosjektet vil bli ferdig i tide?

2.2 Kan du si noe om de siste endringene i prosjektet?

3. Oppgavene som hører til user historien *SCM and General Messages* er alle ferdig. Flytt disse oppgavekortene til kolonnen *done*.

3.1 Hvordan opplevde du denne prosessen?

3.2 Hvis testpersonen ikke ser at informasjonsskjermene oppdateres så si: *Systemet oppdaterte informasjonen på informasjonsskjermene til høyre. Hvordan tolker du denne informasjonen nå?*

3.3 Noe du savner? Noe du synes er unødvendig?

4. Hvor mange brukerhistorier er det?

4.1 Hvor mange er ferdige?

4.2 Naviger til de resterende brukerhistoriene.

4.2.1 Hvordan oppfattes dette?

5. Hvis du ble bedt om å sette deg inn i prosjektet som vises på informasjonsradiator, hvordan ville du da gjort dette?

6. Er det noe forbedringer eller utvidelser du ønsker ved systemet?

Vedlegg F Mal for brukertest - Iterasjon 3

Iterasjon 2 - Brukertesting av Verdandi

Testpersondetaljer: Navn: _____ Alder: _____ Kjønn: _____ Yrke: _____
Antall år i bransjen: _____

Har du benyttet brukerhistorier og informasjonsradiatorer/scrumvegg i tidligere prosjekter. Hvis ja, hvor mange prosjekter?:

Leses til deltaker før vi begynner:

Takk for at du har tatt deg tid til å komme her i dag. Vi skal i dag teste en digital versjon av en informasjonsradiator, også kjent som en *scrumvegg* eller *oppgavetavle*, for å se hvordan denne fungerer i forhold til en fysisk informasjonsradiator.

Vi vil understreke at det er viktig for oss er å få tilbakemelding på er de ulike komponentene til systemet og de handlingene som kan utføres. Vi ønsker derfor at du tenker høyt under hele testen. Det vil si at: hvis det er noe du lurer på, ikke forstår eller hvis du har noe på hjerte så si dette høyt.

Du vil få spørsmål og oppgaver underveis i denne testen.

Utførelsen av disse oppgavene vil bli filmet slik at vi senere kan gå tilbake for å analysere eventuelle problemer med dette systemet. Din identitet vil holdes anonym i hele studien. Du vil få instruksjoner om hva du skal gjøre før hver oppgave. Hvis noe er uklart eller om du ønsker å avbryte testen er det bare å si i fra til meg. Vi stopper kanskje testen underveis.

Intervju

1. Hva er din kjennskap til begrepet informasjonsradiator /scrumvegg?

1.1 Kan du fortelle hvordan du har brukt informasjonsradiatorer / scrumvegger tidligere?

1.1.1 Hvordan så deres informasjonsradiator ut i layout, fargebruk, notasjon?

Kommentar: ha et ark tilgjengelig for å tegne på hvis det er nødvendig

1.1.2 Var det noe du følte var mindre bra eller du følte du manglet?

2. Kjenner du til begrepet brukerhistoriekort/storycards og oppgavekort?

2.1 Har du brukt en informasjonsradiator sammen med brukerhistoriekort ?

2.2 Har du brukt en informasjonsradiator sammen med oppgavekort ?

2.3 Har du brukt en informasjonsradiator med både brukerhistoriekort og oppgavekort? Var det i så fall en struktur som viste hvilke oppgavekort som hørte til hvilke brukerhistoriekort?

3. Kjenner du til begrepet burndowndiagram (burndown chart)? Forklar gjerne hva du forbinder med det begrepet.

3.1 Har du brukt et slik diagram tidligere?

3.2 Vet du hva velocity er?

4. Har du brukt/spilt kinect tidligere?

4.1 Prøvd en gang, flere ganger, ofte...evt annet

Opplæring med kinect

Leses til testpersonen: Før vi begynner testen av selve systemet skal vi gi litt opplæring i bruk av kinect. I opplæringen er målet at du blir kjent med interaksjonsmetoden.

Opplæring

1. Vis kalibreringsposituren utenfor kinectkameraets synsvinkel.

2. Vis hvordan musepekeren velges, høyre hånd opp

3. Vis hvordan man styrer musen med håndflaten

4. Vis hvordan man velger et element med pingvinklikk, venstre arm ut 20 cm.

5. Vis hvordan man gir fra seg kontroll, venstre arm opp over hode.

6. La brukeren prøve stor blinktest 3 ganger. Prøv å få brukeren til å bevege siktet litt rundt

på skjermene slik at interaksjonsmetoden er forstått.

Poengresultat av stor blinktest. Gjør testpersonen oppmerksom på at:

Målet med denne testen er at du skal bli kjent med interaksjonsmetoden, forsøk derfor å bevege siktet rundt på skjermen slik at du får kjenskap til interaksjonsmetoden. Når du føler at du har kontroll; plasser siktet så sentrert som mulig over blinken før du slipper siktet. En poengsum vil dukke opp på skjermen.

Runde 1: X: Y:

Runde 2: X: Y:

Runde 3: X: Y:

Test av Verdandi

Leses til testpersonen: Vi skal nå begynne å teste vår digitale informasjonsradiator. Vi vil i denne fasen gi deg ulike spørsmål og oppgaver. Vi ber deg tenke høyt under hele prosessen, slik at vi forstår ditt synspunkt.

Informasjonsradiatoren viser utviklingen av et gitt system og du skal nå orientere deg i dette prosjektet. Systemet viser et prosjekt som er kommet et stykke i utviklingsprosessen, det er derfor definert brukerhistorier og oppgaver. Åtte personer jobber på prosjektet, inkludert deg.

1. Hva er førsteinntrykket av systemet slik som du ser det nå? Hvordan tolker du innholdet?

1.1 Kan du prøve å forklare strukturen i systemet. Hva er de ulike delene?

1.2 Hvordan tolker du inndelingen av brukerhistoriekort og oppgavekort?

1.2.1 Hvordan tolker du koblingen mellom brukerhistoriekort og oppgavekort?

1.2.2 Hvordan tolker du de ulike fargene på oppgavene?

2. Det er generell prosjektinformasjon på skjermene til høyre. Hvordan er framgangen i prosjektet?

2.1 Ser det ut til at sprinten vil bli ferdig i tide?

2.1.1 Hva er de ulike grafene/linjene i burndowndiagrammet?

2.1.2 Hvor mange poeng gjenstår?

2.1.3 Hvor mange dager gjenstår?

2.2 Kan du si noe om de siste endringene i prosjektet?

2.3 Kan du si noe om kunnskapen til de ansatte?

Kommentar: Kalibrering av kinect gjøres på dette stadiet.

3. Oppgavene som hører til user historien *SCM and General Messages* er alle ferdig. Flytt disse oppgavekortene til kolonnen *done*.

3.1 Hvordan opplevde du denne prosessen?

3.2 Hvis testpersonen ikke ser at informasjonsskjermene oppdateres så si: *Systemet oppdaterte informasjonen på informasjonsskjermene til høyre. Hvordan tolker du denne informasjonen nå?*

3.3 Er det noe som ikke er tydelig? Eventuelt noe som kunne vært tydeligere?

Observatør: vis hvordan man tar fram navigeringsmenyen (venstre arm push fra skulder mot skjerm)

4. Hvor mange brukerhistorier er det?

4.1 Hvor mange er ferdige? //vi er her ute etter omdeltageren ser statusboksen nederst på skjerm 16

Vis navigasjonsgest

4.2 Naviger til de resterende brukerhistoriene.

4.2.1Hvordan oppfattes dette?

5. Er det noe annet du vil si i forhold til systemet? Tilbakemeldinger som ikke er blitt diskutert?

Vedlegg G Sammenligning av kognitive dimensjoner

Tabell 5: viser en sammenligning av de kognitive dimensjonene ved fysiske informasjonsradiatorer og Verdandi. Oppsummeringene av kategoriene på fysiske informasjonsradiatorer er hentet fra Sharp et al. (2009).

Dimensjon	Fysisk informasjonsradiator	Verdandi
Viskositet	<ul style="list-style-type: none"> + Fleksible annoteringer + Lett å endre + Ingen avhengigheter + Enkelt å flytte kort på veggen - Flytting av kort kan føre til konsekvensviskositet - Endre notasjon kan medføre repetisjonsarbeid 	<ul style="list-style-type: none"> + Automatisk sortering + Ingen konsekvensviskositet - Vanskelig å gjøre endringer i oppsettet - Viskose annoteringer - Ikke mulig å flytte flere kort samtidig
Synlighet	<ul style="list-style-type: none"> + Generelt høy synlighet - Applikasjonen er ikke synlig 	<ul style="list-style-type: none"> + Generelt høy synlighet + Synliggjøring av hvem som jobber med hva + Commitmeldinger synliggjør applikasjonen - Kun mulig å se én sprint, ikke hele prosjektet - Begrenset synlighet ift. brukerhistorier
For tidlige forpliktelser	<ul style="list-style-type: none"> + Ingen 	<ul style="list-style-type: none"> - Ikke mulig å reestimere etter hvert steg - Brukerhistorie flyttes til bunnen før brukeren ber om det
Skjulte avhengigheter	<ul style="list-style-type: none"> + Informasjonsradiatoren viser kortenes avhengigheter ift. prosess og applikasjon + Avhengigheter mellom oppgaver og brukerhistorier kan vises - Avhengigheter mellom kort er skjult 	<ul style="list-style-type: none"> + Informasjonsradiatoren viser kortenes avhengigheter ift. prosess og applikasjon + Avhengigheter mellom oppgaver og brukerhistorier vises + Avhengigheter mellom oppgaver kan vises automatisk + Kan forhindre skjulte avhengigheter mellom ulike system - Avhengigheter mellom kort er skjult - Skjulte avhengigheter mellom oppgavekort og commitmeldinger

Uttrykksfulle roller	<ul style="list-style-type: none"> + Uttrykkskraften til brukerhistorier kan økes ved fargeannotering + Uttrykkskraft i forhold til prosess - Lav uttrykkskraft til brukerhistorier 	<ul style="list-style-type: none"> + Notasjonsforskjell mellom brukerhistorier og oppgaver + Fargeannotering på oppgavekort gir økt uttrykkskraft + Forskjellig layout på brukerhistorier og oppgaver gir økt uttrykkskraft + Uttrykkskraft ift. prosess
Utsatt for feil	<ul style="list-style-type: none"> - Informasjonsradiatoren har ingen sikkerhetsmekanismer - Kortene har ingen sikkerhetsmekanismer - Kortene kan falle ned og mistes 	<ul style="list-style-type: none"> + Sikkerhetsmekanismer ift. struktur, fargebruk, annoteringer o.l. + Ikke mulig å flytte oppgavekort til feil brukerhistorie + Sikkerhetsmekanismer ift. sortering av brukerhistorier og oppgaver + Korrekt oppdatert burndowndiagram + Korrekt oppdatert statistikk + Kortene kan ikke mistes og falle ned - Systemfeil kan oppstå
Abstraksjon	<ul style="list-style-type: none"> - Informasjonsradiatoren gir kun enkle abstraksjoner 	<ul style="list-style-type: none"> + Mulighet for å automatisere abstraksjoner - Ingen konfigurasjonsmessig abstraksjon
Sekundærnotasjon	<ul style="list-style-type: none"> + Mulighet for sekundærnotasjon 	<ul style="list-style-type: none"> - Kun tekstlig sekundærnotasjon
Relasjonsnærhet	<ul style="list-style-type: none"> + Relasjonsnærhet mellom brukerhistorier og problemdomenet + Prosessnotasjon er nært relatert til den faktiske prosessen - Ingen relasjonsnærhet til applikasjonen 	<ul style="list-style-type: none"> + Relasjonsnærhet mellom brukerhistorier og problemdomenet + Prosessnotasjon er nært relatert til den faktiske prosessen + Noe relasjonsnærhet til applikasjonen
Konsistent	<ul style="list-style-type: none"> + Cohns mal er konsistent + Når annoteringene er lært, er de enkle å forstå - Vanskelig å forstå for utenforstående - Kan variere fra team til team 	<ul style="list-style-type: none"> + Konsistent struktur - Ikke konsistent bruk av farger
Diffusitet	<ul style="list-style-type: none"> + Konsis notasjon brukt på kortene + Naturlig språk vurderes som konsist - Avhengig av forfatterens formuleringsevne 	<ul style="list-style-type: none"> + Konsis notasjon + Burndowngraf er konsist - Noen store elementer

Vanskelige mentale operasjoner	<ul style="list-style-type: none"> + Krever lite i forhold til prosessen - Vanskelig å forstå applikasjonen - Skjulte avhengigheter mellom kort 	<ul style="list-style-type: none"> + Krever lite i forhold til prosessen + Burndowndiagram gjør det enklere å vurdere prosessen + Lett å koble oppgaver til brukerhistorier - Vanskelig å forstå applikasjonen - Vanskelig å koble commitmeldinger til oppgaver
Midlertidighet	<ul style="list-style-type: none"> + Svært fleksible kort og informasjonsradiator 	<ul style="list-style-type: none"> + Mulig å gjøre «hva om handlinger» - Ikke mulig å endre notasjonen - Informasjonsradiatoren og kortene har mistet noe av sin fleksible natur - Fast struktur, begrensninger i layout
Progressiv evaluering	<ul style="list-style-type: none"> + Viser prosessen - Sier ingen ting om hvor «riktige» brukerhistoriene er - Kortene støtter ikke progressiv evaluering 	<ul style="list-style-type: none"> + Ferdigstillingsprosent på brukerhistoriekort + Burndowndiagram + Gjenstående dager og brukerhistorier

