Usability Feedback i Systemudvikling: Objektorienteret Analyse som Analytisk Mellemled Mellem Usability Testing og Software Testing

Asbjørn Bach Lorentzen
Aalborg Universitet
Informationsvidenskab Cand.IT
Kandidatspeciale

Titel: Usability feedback i systemudvikling. Objektorienteret analyse som analytisk mellemled mellem usability testing og software testing

Asbjørn Bach Lorentzen
Studienr: 20192835

Antal tegn: 118.473
Antal normalsider: 49.36

Vejleder: David Jakobsen

Abstract

Digital system development requires attention not only to technical correctness but also to usability. Systems may satisfy functional and non-functional requirements while still creating difficulties for users in practice. Usability testing addresses this challenge by providing insight into how users interact with systems in specific contexts. However, usability feedback is often qualitative, context-dependent, and interpretative, which makes it difficult to integrate directly into structured system development processes. In contrast, software testing is typically based on verification, predefined requirements, and measurable outcomes. This creates a structural tension between usability testing and software testing, as the two approaches rely on different test logics and produce different forms of knowledge.

This thesis investigates how usability feedback can be analyzed and structured so that usability knowledge can contribute more directly to system development and decision-making. The thesis adopts a theoretical and analytical approach and does not include empirical user studies. Instead, the analysis is based on existing literature and selected usability evaluation cases related to usability testing, software testing, object-oriented analysis and design, ontological structuring, and relational knowledge representations.

The analysis shows that usability feedback cannot be understood solely as isolated interface problems or individual user experiences, but also as knowledge about relationships between users, system functions, workflows, information structures, and contextual conditions. The thesis argues that usability problems in complex systems often emerge through structural and organizational relationships rather than through individual functions in isolation.

The central contribution of the thesis is the introduction of object-oriented analysis as an analytical intermediary between usability testing and system development. By applying concepts such as objects, responsibilities, and relationships, usability feedback can be analyzed in relation to the structural organization of systems. Ontological structuring and knowledge graph-inspired relational perspectives further support the organization of usability feedback as interconnected analytical relationships rather than isolated observations.

The thesis concludes that usability feedback can support more systematic analysis and design work when it is structured through object-oriented, ontological, and relational perspectives. Rather than functioning as a new development method, the proposed approach should be understood as an analytical perspective that supports reflection, interpretation, and structural understanding in system development.

Indhold

- Abstract
- Indhold

Indledning

1. Problemfelt

2. Litteraturreview
    - 2.1 Litteratursøgning og afgrænsning
    - 2.2 Usability testing
    - 2.3 Software testing
    - 2.4 Objektorienteret analyse og design

3. Metode
    - 3.1 Metodisk tilgang
    - 3.2 Teoretisk design af usability tests og cases
    - 3.3 Analytisk ramme
    - 3.4 Metodiske afgrænsninger

4. Teori
    - 4.1 Usability og usability engineering
        - 4.1.1 Begrebet usability
        - 4.1.2 Usability som analytisk ramme
        - 4.1.3 Usability som en del af systemkvalitet
        - 4.1.4 Begrænsninger ved usability
    - 4.2 Usability testing som metode
        - 4.2.1 Formålet med usability tests
        - 4.2.2 Kvalitativ vs kvantitativ testlogik
        - 4.2.3 Den fortolkningsafhængige karakter af usability tests
        - 4.2.4 Begrænsninger ved testing
    - 4.3 Software testing og teststandarder
        - 4.3.1 Software testing som verificerende praksis
        - 4.3.2 Testniveauer og testmål
        - 4.3.3 Testprocesser, dokumentation og sporbarhed
        - 4.3.4 Usability som non-funktionel kvalitet
        - 4.3.5 Spændingsfelt mellem usability testing og software testing
    - 4.4 Objektorienteret analyse og design som strukturel tænkning
        - 4.4.1 Behovet for strukturel tænkning i systemudvikling
        - 4.4.2 Objektorienteret analyse som analytisk tilgang
        - 4.4.3 Objekter, ansvar, relationer
        - 4.4.4 Fra problemforståelse til strukturel model
        - 4.4.5 Objektorientering som bro mellem usability og systemudvikling
    - 4.5 Ontologi og strukturering af viden
        - 4.5.1 Ontologi som begrebsramme
        - 4.5.2 Knowledge Graphs
        - 4.5.3 Analytisk strukturering af usability feedback
            - 4.5.3.1 Ontologisk identifikation af begreber og relationer

4.5.3.2 Relationel strukturering gennem knowledge graphs
4.5.3.3 Objektorienteret fortolkning af usability feedback
4.5.3.4 Sammenkobling af de analytiske perspektiver
5. Analyse
5.1 Karakteren af usability feedback
5.1.1 Usability feedback som kontekstafhængig viden
5.1.2 Fortolkningsbehov og usikkerhed
5.1.3 Delkonklusion: usability feedback som analytisk udfordring
5.2 Spændingsfeltet mellem usability testing og software testing
5.2.1 Software testings verificerende logik
5.2.2 Usability testings diagnostiske logik
5.2.3 Analytisk kontrast mellem testlogikkerne
5.2.4 Konsekvenser i praksis og behovet for et mellemled
5.3 Objektorienteret analyse som strukturel oversættelse
5.3.1 Objekter som analytiske enheder
5.3.2 Ansvar som fortolkningsramme
5.3.3 Relationer som struktur for kompleksitet
5.4 Fra indsigt til strukturel model
5.4.1 Feedback som problemforståelse
5.4.2 Fra problemforståelse til strukturel repræsentation
5.4.3 Strukturelle modeller som analytisk arbejdsredskab
5.5 Analytiske cases
5.5.1 Case 1 - Usability Evaluation of Published Clinical guidelines on the Web: A Case Study
5.5.1.1 Relationelle usability problemer i informationsarkitekturen
5.5.1.2 Ontologisk og relationel strukturering af usability feedback
5.5.1.3 Objektorienteret fortolkning af usability-problemerne
5.5.1.4 Delkonklusion
5.5.2 Case 2 - Evaluating usability of a commercial electronic health record: A case study
5.5.2.1 Workflow som usability-kontekst
5.5.2.2 Strukturel analyse af ansvar og relationer
5.5.2.3 Fra usability-problem til organisatorisk struktur
5.5.2.4 Delkonklusion
5.5.3 Tværgående analytiske pointer fra casene
5.6 Konsekvenser for design og analyse af usability testing
5.6.1 Skift i testformål
5.6.2 Konsekvenser for testopgaver
5.6.3 Konsekvenser for observation og analyse
5.6.4 Usability testing som strukturerende redskab
6. Diskussion
6.1 Diskussion af specialets analytiske bidrag
6.2 Usability feedback som strukturel og relationel viden
6.3 Objektorienteret analyse som analytisk mellemled

6.4 Begrænsninger og analytiske spændinger
6.5 Implikationer for praksis
6.6 Perspektiver for fremtidigt arbejde
7. Konklusion
Litteraturliste

Indledning

Udviklingen af digitale systemer stiller krav til både teknisk kvalitet og høj anvendelighed. Et system kan opfylde tekniske krav og samtidig skabe problemer for brugerne i praksis. Derfor spiller brugertests en central rolle i systemudvikling, da de giver indsigt i, hvordan brugere interagerer med et system, og hvilke udfordringer der opstår. Usability testing producerer en type viden, der ofte er kvalitativ, kontekstafhængig og fortolkningsbaseret. Selvom denne viden er vigtig for forståelsen af systemets anvendelighed, kan den være vanskelig at omsætte til konkrete beslutninger i systemudviklingen. Samtidig er mange udviklingsprocesser præget af software testing, som i højere grad bygger på kravspecifikationer, målinger og verificering. Dette skaber et spændingsfelt mellem usability testing og software testing, hvor usability feedback risikerer at få begrænset betydning i analyse- og designarbejdet.

Specialet tager udgangspunkt i dette spændingsfelt og undersøger, hvordan usability feedback kan analyseres og struktureres, så usability-viden i højere grad kan indgå i kvalificerede systembeslutninger. Med afsæt i usability testing, software testing samt objektorienteret analyse og design udforsker specialet, hvordan begreber som objekter, ansvar og relationer kan anvendes som analytisk ramme til at forbinde brugerindsigter med systemets struktur. Specialet inddrager desuden ontologiske perspektiver og knowledge graphs som tilgange til at organisere og repræsentere usability-viden med henblik på at understøtte en mere systematisk forståelse af systemets problemområder.

Specialet har et teoretisk og analytisk fokus og baserer sig ikke på egne empiriske undersøgelser, men anvender eksisterende usability cases som analytiske eksempler. Gennem analyser af forskellige usability evaluations undersøger dette speciale, hvordan usability testing frembringer fortolkningsafhængig viden, og hvordan denne viden kan være vanskelig at integrere direkte i systemudviklingens mere strukturerede og verificerende processer. På denne baggrund undersøger specialet, hvordan objektorienteret analyse, ontologisk strukturering og relationelle modeller kan fungere som analytiske mellemled mellem usability testing og software testing. Formålet med specialet er at bidrage til en

bedre forståelse af, hvordan usability-viden kan analyseres og struktureres som grundlag for kvalificerede systembeslutninger i systemudvikling.

1. Problemfelt

Kvaliteten af digitale systemer afhænger i høj grad af den testning og evaluering, der gennemføres forud for systemets implementering. I takt med at digitale løsninger bliver mere komplekse og anvendes i varierede organisatoriske og sociale sammenhænge, stilles der stigende krav til, at systemer ikke blot er teknisk korrekte, men også brugbare og understøttende for brugernes behov og arbejdspraksisser. Manglende fokus på brugervenlighed kan medføre ineffektiv systemanvendelse, frustration blandt brugerne og i sidste ende manglende accept af løsningen.

I denne sammenhæng spiller brugertests, herunder usability testing, en central rolle, idet de skaber indsigt i, hvordan brugere reelt interagerer med et system, og hvilke udfordringer der opstår. Gennem observation af brugeres opgaveløsning kan usability testing producere viden om brugerens møde med systemet, som ikke nødvendigvis identificeres gennem tests af systemets tekniske eller funktionelle egenskaber. Usability feedback består ofte af kvalitative og situationsafhængige indsigter, som kræver analytisk fortolkning (Hertzum & Clemmensen, 2012). Forskellige brugere, evaluatorer og organisatoriske sammenhænge kan derfor føre til forskellige forståelser af det samme systems usability. Dette gør usability feedback vanskelig at omsætte direkte til konkrete systemkrav eller tekniske ændringer.

Inden for softwareudvikling eksisterer samtidig etablerede teststandarder og rammeværk, såsom International Software Testing Qualifications Board (ISTQB), der har til formål at sikre systemkvalitet gennem strukturerede, kravbaserede og verificerende testprocesser (Stapp et al., 2024). Disse tilgange fokuserer primært på at dokumentere og verificere, hvorvidt et system lever op til specificerede krav, og producerer dermed testresultater, der er målbare og direkte anvendelige i den tekniske videreudvikling af systemet. Brugerens oplevelse og anvendelse af systemet indtager her en mere indirekte rolle.

Der opstår således et spændingsfelt mellem usability testing og software testing, idet de to testtilgange producerer forskellige typer viden med forskellige formål og kvalitetsforståelser. Hvor software testing søger at verificere, om et system fungerer som specificeret, søger usability testing i højere grad at skabe indsigt i, hvordan systemet opleves og anvendes i praksis. Denne forskel kan medføre, at usability feedback får vanskelige vilkår i systemudviklingsprocessen, fordi feedbacken ikke umiddelbart kan omsættes til krav, testcases eller tekniske ændringer. Konsekvensen kan være, at usability-relaterede indsigter behandles fragmenteret, sent i udviklingsprocessen eller reduceres til overfladiske justeringer af brugergrænsefladen.

Objektorienteret analyse og design udgør i denne sammenhæng en udbredt tilgang til systemudvikling, der tilbyder principper for strukturering af komplekse systemer gennem begreber som objekter, ansvar og relationer (Mathiassen et al., 2018). Disse principper er udviklet med henblik på at håndtere kompleksitet og forandring over tid og kan anvendes som en analytisk ramme til at strukturere og organisere usability-viden. Objektorienteret analyse arbejder med relationer mellem aktører, handlinger, systemfunktioner og domænebegreber og åbner dermed mulighed for at forstå usability feedback som mere end enkeltstående problemer eller observationer.

Samtidig peger dette på behovet for begrebsmæssige og relationelle strukturer, der kan understøtte organiseringen af usability-viden på tværs af brugssituationer, observationer og systemfunktioner. Ontologiske perspektiver og relationelle modeller kan i denne sammenhæng bidrage med analytiske principper for strukturering af komplekse sammenhænge og skabe forbindelser mellem usability feedback, systemfunktioner og udviklingsmæssige beslutninger. På baggrund af dette problemfelt opstilles følgende problemformulering:

Hvordan kan usability feedback fra brugertests analyseres og struktureres, så usability-viden kan omsættes til kvalificerede systembeslutninger, og hvilken rolle kan objektorienteret analyse og design spille som bindeled mellem usability testing og software testing?

I dette speciale forstås systembeslutninger som analytiske og designmæssige beslutninger vedrørende systemets struktur, funktioner, informationsorganisering og understøttelse af arbejdspraksisser. Begrebet omfatter dermed ikke alene tekniske implementeringsbeslutninger, men også beslutninger om relationer mellem systemets elementer, workflows og brugerinteraktioner.

2. Litteraturreview

2.1 Litteratursøgning og afgrænsning

Som led i udarbejdelsen af specialets litteraturgrundlag er der gennemført litteratursøgninger for at identificere eksisterende forskning om usability testing, software testing og objektorienteret analyse og design. Søgningerne blev gennemført via aub.aau.dk og Google Scholar med forskellige kombinationer af søgeord på både dansk og engelsk, herunder “usability”, “usability testing”, “user-centered design”, “object-oriented analysis” og “usability evaluation case study”. Der blev anvendt både kombinerede og bredere søgninger, hvor begreberne indgik hver for sig.

Formålet med litteratursøgningerne var dels at identificere forskning, der kunne belyse spændingsfeltet mellem usability testing og software testing, dels at undersøge, hvorvidt der eksisterer litteratur, som eksplicit forbinder usability feedback med objektorienteret analyse og design. Søgningerne viste, at forskning om usability testing primært fokuserer på brugeroplevelse, evaluering og interaktion, mens litteratur om objektorienteret analyse og design hovedsageligt retter sig mod strukturering, modellering og systemudvikling. Der blev kun i begrænset omfang identificeret forskning, der direkte undersøger usability feedback som input til objektorienteret analyse og design. På denne baggrund er litteraturgrundlaget afgrænset til centrale værker, der repræsenterer specialets væsentligste teoretiske perspektiver: usability testing og usability evaluation (Hertzum, 2016; Hertzum & Clemmensen, 2012; Hertzum et al., 2009), software testing (Stapp et al., 2024) samt objektorienteret analyse og design (Mathiassen et al., 2018).

Ontologiske perspektiver og knowledge graphs indgår desuden som supplerende teoretiske perspektiver i specialet. Disse perspektiver blev introduceret gennem litteratur fra undervisningsforløb og efterfølgende understøttet af udvalgte værker om ontologi, knowledge graphs og relationelle vidensstrukturer (Guarino et al., 2009; Hogan et al., 2021; Jakobsen & Graf, 2022; Li et al., 2024).

Derudover blev litteratursøgningerne anvendt til at identificere usability cases, der kunne fungere som analytiske cases i specialet. Disse cases blev blandt andet identificeret gennem søgninger som “usability evaluation case study” i Google Scholar. Formålet med disse cases er ikke at gennemføre empiriske evalueringer, men at anvende eksisterende usability evaluations som analytiske eksempler på komplekse usability-problemer og deres relation til systemstruktur og arbejdspraksis. Specialets bidrag ligger således ikke i at etablere en ny

usability-metode eller i at gentage en eksisterende forskningsretning, men i analytisk at forbinde usability testing, objektorienteret analyse og relationelle struktureringsperspektiver med henblik på at undersøge, hvordan usability feedback kan analyseres og struktureres, så usability-viden kan indgå i mere kvalificerede systembeslutninger.

2.2 Usability testing

Som teoretisk fundament for forståelsen af usability testing anvendes Hertzum og Clemmensen (2012), der repræsenterer en nyere og mere kontekstorienteret forståelse af usability. Forfatterne beskriver usability som et begreb, der forstås og fortolkes forskelligt afhængigt af organisatoriske mål, aktører og anvendelsessituationer (Hertzum & Clemmensen, 2012). Usability testing bliver dermed ikke alene en metode til at identificere problemer i brugerens interaktion med et system, men også en praksis, der producerer fortolknings- og kontekstafhængig viden om brugerens møde med systemet. Denne forståelse er central for specialet, fordi usability feedback ikke betragtes som objektive fejl eller entydige problemer, men som indsigter, der skal analyseres og fortolkes i relation til systemets funktioner, brugssituationer og organisatoriske kontekst.

Hertzum, Hansen og Andersen (2009) bidrager desuden med et metodisk perspektiv på usability evaluation ved at undersøge, hvordan think-aloud-protokoller kan påvirke både brugeradfærd og mental workload under usability tests (Hertzum et al., 2009). Artiklen understøtter dermed forståelsen af usability testing som en situationel og påvirkelig evalueringspraksis, hvor de identificerede usability-problemer ikke kan betragtes som fuldstændigt neutrale eller objektive observationer.

Hertzum (2016) inddrages desuden som perspektiv på usability testings metodiske udfordringer og begrænsninger. Hertzum diskuterer blandt andet, hvordan usability testing påvirkes af forhold som testens timing, anvendelsen af think-aloud samt fokus på problemidentifikation frem for organisatoriske effekter (Hertzum, 2016). Artiklen understøtter specialets fokus på usability testing som en analytisk og fortolkningsafhængig praksis frem for en metode, der producerer entydige eller objektive resultater.

2.3 Software testing

For forståelsen af software testing anvendes *ISTQB® Certified Tester Foundation Level: A Self-Study Guide Syllabus v4.0* skrevet af Stapp et al. (2024), der repræsenterer

International Software Testing Qualifications Boards (ISTQB) Foundation Level Syllabus. Kilden anvendes som reference for software testingens centrale begreber, testlogikker og kvalitetsforståelser i praksis. ISTQB-materialet repræsenterer en internationalt anerkendt standard inden for software testing og anvendes bredt i både industri og uddannelse. I dette speciale anvendes Stapp et al. (2024) ikke som teoretisk perspektiv på brugeroplevelse, men som repræsentation af den kravbaserede og verificerende testlogik, der præger software testing i systemudviklingsprojekter.

Kilden er valgt, fordi den eksplicit beskriver software testing som en praksis baseret på kravspecifikationer, testcases, coverage og sporbarhed. Denne forståelse er central for specialets analyse af spændingsfeltet mellem usability testing og software testing, da den illustrerer, hvordan kvalitet inden for software testing primært forstås som opfyldelse af specificerede funktionelle og non-funktionelle krav. Anvendelsen af ISTQB-litteratur muliggør dermed en analyse af software testing, som den praktiseres snarere end som et abstrakt teoretisk ideal.

2.4 Objektorienteret analyse og design

Som teoretisk grundlag for den objektorienterede tilgang anvendes *Object Oriented Analysis & Design* af Mathiassen et al. (2018), der præsenterer objektorienteret analyse og design som et principbaseret rammeværk til strukturering af komplekse problemområder i systemudvikling. Mathiassen et al. adskiller sig fra mere teknisk orienteret litteratur ved at fremhæve objektorientering som en strukturel måde at forstå systemer på frem for som en implementerings- eller programmeringsrelateret tilgang.

Begreber som objekter, ansvar og relationer anvendes til at analysere og modellere systemers problemområder og kontekst før implementering. Denne forståelse gør værket særligt relevant for specialets formål. I dette speciale anvendes objektorienteret analyse ikke med henblik på kodning eller design af softwarearkitektur, men som et analytisk redskab til at strukturere usability feedback. Mathiassen et al. (2018) er valgt, fordi værket tilbyder et strukturelt begrebsapparat, der kan anvendes til at oversætte usability-observationer til problemstillinger i systemudvikling.

3. Metode

Dette speciale har til formål at udvikle en teoretisk og metodisk forståelse af, hvordan usability-viden fra brugertests kan analyseres og struktureres, så den kan indgå i kvalificerede systembeslutninger i systemudvikling. Specialet baserer sig ikke på egen empirisk datagenerering, men på en kvalitativ, teoretisk og analytisk tilgang, hvor eksisterende litteratur, usability cases og metodiske perspektiver inden for usability testing, software testing og objektorienteret analyse og design analyseres og diskuteres. Den metodiske tilgang er abduktiv, idet specialet bevæger sig iterativt mellem teori og analyse. På baggrund af eksisterende teorier, cases og metodiske perspektiver identificeres centrale problemstillinger og begrebslige spændinger, som danner grundlag for en analytisk undersøgelse af usability testing og usability feedback. Formålet er ikke at validere teorier empirisk, men at undersøge og diskutere, hvordan forskellige metodiske og analytiske perspektiver kan kombineres for at understøtte struktureringen af usability-viden i systemudvikling.

3.1 Metodisk tilgang

Metodisk baserer specialet sig på en analytisk og sammenlignende tilgang, hvor usability testing og software testing analyseres som to forskellige testlogikker med henblik på at identificere deres respektive vidensformer, begrænsninger og integrationsproblemer. Specialet anvender en kvalitativ forskningsstrategi med fokus på analytisk fortolkning og begrebsudvikling. Den kvalitative tilgang muliggør en dybdegående undersøgelse af de teoretiske og metodiske antagelser, der danner grundlag for usability testing, samt hvordan disse relaterer sig til praksisser i systemudvikling.

Usability testing anvendes ikke i specialet som empirisk metode, men som et analytisk undersøgelsesfelt. Med udgangspunkt i nyere usability-forskning forstås usability som en kontekst- og fortolkningsafhængig kvalitetsdimension, hvor brugerens oplevelse af systemet formes gennem relationer mellem bruger, system og anvendelsessituation (Hertzum & Clemmensen, 2012). Fokus rettes derfor ikke mod usability som et stabilt og målbart kvalitetskriterium, men mod hvordan usability-viden produceres, fortolkes og anvendes i praksis. Denne forståelse gør det muligt at analysere usability feedback som mere end enkeltstående problemer eller observationer og i stedet undersøge, hvordan usability-indsigter kan struktureres og relateres til systemudviklingens beslutningsprocesser.

3.2 Teoretisk design af usability tests og cases

Som del af analysen opstilles et teoretisk funderet design for usability tests, der anvendes som analytisk ramme til at diskutere, hvordan usability-viden fra brugertests kan struktureres og anvendes i systemudvikling. Testdesignet fungerer ikke som en praktisk opskrift eller et forslag til implementering, men som et analytisk eksempel, der muliggør en systematisk diskussion af usability testing og den type viden, som testformen frembringer.

Med udgangspunkt i usability-litteraturen analyseres centrale komponenter, der traditionelt indgår i usability testing. Følgende elementer anvendes som analytiske komponenter i forståelsen af usability tests:

- udvælgelse af testdeltagere, hvor fokus er på repræsentativitet i relation til brugskontekst frem for statistisk generalisering
- formulering af testopgaver som analytiske konstruktioner, der afspejler centrale brugerhandlinger og anvendelsessituationer
- observation som princip til identifikation af mønstre og udfordringer i brugerens interaktion med systemet
- think-aloud-metoden som eksempel på den refleksive og fortolkningskrævende viden, der ofte produceres i usability testing (Hertzum et al., 2009)

Disse elementer diskuteres og analyseres teoretisk uden at blive empirisk gennemført. Formålet er at undersøge, hvordan usability tests kan struktureres analytisk, og hvordan den viden, der frembringes gennem usability testing, kan bruges som input til systemudvikling.

Specialet inddrager desuden eksisterende usability case studies som analytiske eksempler på, hvordan usability feedback produceres, fortolkes og anvendes i praksis. Casesene anvendes ikke som empirisk datagrundlag i traditionel forstand, men som refleksive analyseobjekter, der muliggør undersøgelse af usability-videns kontekstafhængighed og fortolkningskarakter. De udvalgte cases anvendes til at analysere, hvordan forskellige aktører, brugssituationer og organisatoriske sammenhænge påvirker forståelsen og anvendelsen af usability feedback.

3.3 Analytisk ramme

For at undersøge udfordringen med at omsætte usability-viden til systemudvikling inddrages objektorienteret analyse og design som en strukturerende ramme. Objektorienterede principper såsom ansvarsfordeling, modularitet og relationer anvendes ikke i teknisk forstand

(Mathiassen et al., 2018), men som strukturelle begreber, der kan understøtte organiseringen af usability-indsigter. Ved at kombinere perspektiver fra usability testing og software engineering bidrager specialet med en teoretisk forståelse af, hvordan usability-viden kan integreres mere systematisk i udviklingsprocessen.

Ontologiske perspektiver og knowledge graphs inddrages desuden som konceptuelle redskaber til at undersøge, hvordan usability-viden kan organiseres gennem begreber og relationer (Guarino et al., 2009; Hogan et al., 2021). Disse anvendes ikke som tekniske implementeringsværktøjer, men som modeller til at strukturere relationer mellem brugere, handlinger, funktioner, problemer og kontekster. Den analytiske ramme anvendes således til at diskutere, hvordan kvalitative og fortolkningsafhængige brugeroplevelser kan struktureres og relateres til systemets strukturelle og organisatoriske sammenhænge uden at reducere usability til isolerede krav eller interfaceproblemer. Denne ramme danner grundlag for den efterfølgende analyse af spændingsfeltet mellem usability testing og software testing.

3.4 Metodiske afgrænsninger

Specialet har ikke til formål at evaluere konkrete systemer eller måle effekten af specifikke usability-metoder i praksis. Resultaterne kan derfor ikke generaliseres empirisk i traditionel forstand, men bidrager i stedet med analytisk og metodisk indsigt. Specialets bidrag ligger i den teoretiske strukturering af usability-viden og i diskussionen af, hvordan usability-indsigter kan forstås, organiseres og omsættes til kvalificerede systembeslutninger i systemudvikling.

4. Teori

4.1 Usability og usability testing

Dette afsnit introducerer centrale begreber og perspektiver inden for usability med fokus på, hvordan usability kan betragtes som en kompleks og kontekstafhængig kvalitetsdimension i interaktive systemer. Formålet er ikke at opstille normative designprincipper, men at etablere et analytisk grundlag for at undersøge, hvilken type viden usability testing producerer, og hvorfor denne viden kan være vanskelig at omsætte til strukturerede beslutninger i systemudvikling. Afsnittet danner det teoretiske udgangspunkt for den efterfølgende analyse af spændingsfeltet mellem usability testing og software testing.

4.1.1 Begrebet usability

Usability kan ikke betragtes som en entydig eller stabil egenskab ved et system, men må forstås som et komplekst og flertydigt begreb, der afhænger af både kontekst, bruger og perspektiv. Studier af usability-professionelles praksis viser, at usability ikke opfattes som én samlet størrelse, men som en sammensætning af forskellige og ofte overlappende forståelser (Hertzum & Clemmensen, 2012, s. 26-27).

Hertzum identificerer flere forskellige “billeder” af usability, herunder situational, organizational og perceived usability, som hver især fremhæver forskellige aspekter af systemets anvendelse og værdi i praksis (Hertzum, 2016, s. 83-84). Disse perspektiver understreger, at usability ikke blot handler om systemets egenskaber, men om forholdet mellem system, bruger og kontekst. Samtidig viser empiriske studier, at usability-professionelle anvender en bred vifte af begreber, som går ud over klassiske definitioner som eksempelvis ISO-standarder. Disse definitioner dækker kun en del af den måde usability faktisk fortolkes og anvendes i praksis (Hertzum & Clemmensen, 2012, s. 26-27).

Usability-evaluering producerer dermed en type viden, hvor vurderingen af et system formes af, hvilke aspekter der fremhæves, og hvordan de relateres til konkrete brugssituationer. Dette har direkte betydning for usability testing, da det indebærer, at de observationer og problemer, der identificeres, ikke er fakta, men fortolkninger af brugerens interaktion med systemet.

4.1.2 Usability som analytisk ramme

I stedet for at forstå usability som et fast sæt af målbare dimensioner kan det være mere hensigtsmæssigt at betragte usability som et analytisk felt bestående af forskellige former for vurderinger. Hertzum og Clemmensen viser, at usability-professionelle typisk arbejder med både utilitaristiske og oplevelsesbaserede aspekter af systembrug (Hertzum & Clemmensen, 2012, s. 32-33). De utilitaristiske aspekter relaterer sig til brugerens evne til at løse opgaver effektivt, mens de oplevelsesbaserede aspekter omhandler brugerens oplevelse, følelser og engagement i brugen af systemet. Dette peger på, at usability i praksis omfatter en bredere forståelse af brugeroplevelse, som ikke kan reduceres til standardiserede målinger eller entydige kriterier.

Denne forståelse betyder, at usability testing ikke blot identificerer isolerede problemer, men producerer viden om relationer mellem bruger, system og kontekst. Usability fungerer dermed som en analytisk ramme, der synliggør, hvilke aspekter af interaktionen der er relevante, uden nødvendigvis at give klare anvisninger på, hvordan disse skal omsættes til konkrete systembeslutninger. Den relationelle og fortolkningsafhængige forståelse af usability peger samtidig på behovet for strukturerede begrebsrammer, der kan understøtte organiseringen og repræsentationen af usability-viden i systemudvikling. Usability feedback kan i denne sammenhæng forstås som relationel viden om sammenhænge mellem brugere, handlinger, funktioner og kontekster frem for som enkeltstående observationer eller fejl.

4.1.3 Usability som en del af systemkvalitet

Usability udgør en central del af systemkvalitet, men adskiller sig fra mange andre kvalitetsattributter ved sin kontekstafhængighed og fortolkningskarakter. Hvor tekniske kvaliteter ofte kan specificeres og verificeres gennem entydige kriterier, er usability tæt knyttet til den konkrete anvendelse af systemet og de praksisser, det indgår i. Hertzum fremhæver, at usability ikke alene kan bruges ud fra laboratoriebaserede tests, men må vurderes i relation til systemets anvendelse i organisatoriske sammenhænge (Hertzum, 2016, s. 84-85). Dette perspektiv betegnes som organizational usability og fokuserer på, hvorvidt systemet understøtter de arbejdsprocesser og mål, det er designet til at indgå i.

Denne forståelse indebærer, at usability ikke kun handler om at undgå problemer, men om at understøtte ønskede effekter i praksis. Hertzum argumenterer for, at usability-evaluering i højere grad bør fokusere på, hvorvidt systemet skaber de ønskede resultater, frem for

udelukkende at identificere fejl (Hertzum, 2016, s. 85-86). Konsekvensen er, at usability bidrager med vigtig indsigt i systemets anvendelse, men samtidig kan være vanskelig at integrere direkte i strukturerede og kravbaserede udviklings- og testprocesser. Dette skaber en forskel mellem usability og mere formelle kvalitetsforståelser, som specialet adresserer videre. Samtidig peger denne forskel på behovet for analytiske og relationelle modeller, der kan understøtte struktureringen af usability-viden og skabe forbindelser mellem brugerindsigter og systemets strukturelle sammenhænge.

4.1.4 Begrænsninger ved usability

Selvom usability tilbyder et nuanceret perspektiv på systemkvalitet, er begrebet forbundet med en række begrænsninger. Usability er ikke en entydig størrelse, men afhænger af, hvilket perspektiv der anvendes. Forskellige brugere kan derfor have forskellige forståelser af, hvad der udgør god usability, hvilket kan føre til forskellige vurderinger af det samme system (Hertzum & Clemmensen, 2012). Samtidig indebærer usability en række modsatrettede hensyn. Et system kan eksempelvis være optimeret til effektiv opgaveløsning, men samtidig opleves som rigidt eller begrænsende. Omvendt kan systemer, der understøtter fleksibilitet og brugeroplevelse, være mindre effektive i forhold til specifikke arbejdsopgaver.

Hertzum understreger desuden, at systemers egenskaber først bliver meningsfulde for brugerne, når de påvirker deres faktiske arbejde og praksis (Hertzum, 2016, s. 84-86). Dette betyder, at usability ikke kan fastlægges fuldt ud på forhånd, men udvikler sig i mødet mellem system og brug. Samlet set betyder dette, at usability ikke kan reduceres til et stabilt sæt af kriterier, men må forstås som en dynamisk og fortolkningsafhængig kvalitetsdimension. Dette skaber samtidig en udfordring i systemudvikling, hvor usability-viden må analyseres og struktureres, før den kan indgå i mere systematiske udviklings- og designbeslutninger.

4.2 Usability testing som metode

Dette afsnit uddyber usability testing som metodisk tilgang med fokus på den type viden, testformen producerer, og de metodiske antagelser, der ligger til grund for denne viden. Formålet er ikke at beskrive konkrete testprocedurer, men at analysere usability testing som en fortolkningsafhængig og kontekstafhængig testform samt at tydeliggøre, hvorfor resultaterne herfra ofte er vanskelige at integrere i systemudviklingens mere strukturerede og verificerende praksisser. I dette speciale forstås usability feedback som de observationer, fortolkninger og indsigter, der opstår gennem analyse af brugeres interaktion med et system

under usability testing. Feedback forstås dermed ikke som objektive fejlrapporter, men som relationel og fortolkningsafhængig viden om sammenhænge mellem bruger, system og kontekst. Afsnittet udgør dermed det teoretiske grundlag for den efterfølgende analyse af spændingsfeltet mellem usability testing og software testing.

4.2.1 Formålet med usability tests

Formålet med usability testing er at opnå indsigt i, hvordan brugere interagerer med et system i praksis, og hvordan systemet understøtter eller hindrer deres opgaveløsning. Traditionelt er usability testing blevet forstået som en metode til at identificere usability-problemer, men nyere forskning peger på, at denne forståelse er utilstrækkelig. Hertzum argumenterer for, at usability ikke bør reduceres til identifikation af problemer, men i stedet bør forstås i relation til de effekter, systemet har i brugssituationer (Hertzum, 2016, s. 85-86). Dette indebærer, at usability testing ikke blot identificerer problemer, men også skaber indsigt i relationer mellem brugere, handlinger, systemfunktioner og kontekster.

Samtidig viser forskning, at usability testing ikke genererer objektive eller entydige resultater. Studier af usability-praksis viser, at forskellige evaluatorer ofte identificerer forskellige forhold i de samme testsituationer, hvilket peger på, at usability findings i høj grad er fortolkningsafhængige (Hertzum & Clemmensen, 2012, s. 27-28). Usability testing kan derfor ikke forstås som en metode, der producerer stabile fakta om et system, men snarere som en metode, der frembringer situeret og fortolkningskrævende viden. Usability feedback kan i denne sammenhæng forstås som relationel viden, hvor betydningen af enkelte observationer afhænger af deres sammenhæng med andre elementer i systemets anvendelseskontekst. Dette har direkte betydning for, hvordan testresultater kan anvendes i systemudvikling.

4.2.2 Kvalitativ vs kvantitativ testlogik

Usability testing adskiller sig fra traditionelle softwaretests ved at være baseret på en kvalitativ testlogik, hvor fokus er på forståelse frem for måling. Hvor kvantitative testmetoder søger at skabe reproducerbare og generaliserbare resultater, har usability testing til formål at skabe indsigt i brugerens interaktion med systemet i konkrete situationer. Denne tilgang indebærer, at usability testing ikke primært fokuserer på, hvor ofte et problem opstår, men på hvordan og hvorfor bestemte udfordringer opstår i relationen mellem bruger, system og kontekst. Resultaterne består derfor ikke af statistiske målinger, men af fortolkninger af brugerens handlinger, oplevelser og interaktioner.

Samtidig er usability testing påvirket af den metode, der anvendes under testen. Hertzum viser eksempelvis, at think-aloud-metoder kan påvirke brugerens adfærd og mentale belastning, hvilket betyder, at testresultaterne ikke kan betragtes som neutrale observationer (Hertzum et al., 2009, s. 165-166). Den kvalitative tilgang betyder dermed, at usability testing frembringer data, der er kontekstafhængige og påvirket af både bruger, evaluator og testopsætning. Dette gør det vanskeligt at anvende resultaterne direkte i strukturerede udviklingsprocesser, da usability-indsigter ofte må fortolkes og organiseres, før de kan omsættes til udviklings- og designbeslutninger.

4.2.3 Den fortolkningsafhængige karakter af usability tests

Usability testing kan karakteriseres som en fortolkningsafhængig testform, hvor resultaterne ikke fremstår som entydige svar, men som beskrivelser af situationer, der kræver analyse. Hvor verificerende testformer søger at afgøre, om et system opfylder fastlagte krav, skaber usability testing i stedet indsigt i, hvordan systemet anvendes i praksis, og hvor denne anvendelse udfordres. Disse indsigter er ikke i sig selv forklaringer eller løsninger, men må fortolkes i relation til systemets struktur og kontekst. Forskning viser, at usability findings ikke er stabile, men varierer afhængigt af, hvem der analyserer data, og hvordan de fortolkes (Hertzum & Clemmensen, 2012, s. 27-28).

Dette betyder, at usability feedback ikke kan omsættes direkte til designbeslutninger, men kræver en analytisk proces, hvor observationer struktureres og sættes i relation til systemets opbygning. Dette peger på behovet for analytiske modeller, der kan organisere usability-indsigter gennem begreber og relationer frem for som isolerede usability-problemer. Denne fortolkningsafhængighed er central for specialets problemstilling, da den understreger behovet for analytiske rammer, der kan understøtte overgangen fra usability-viden til systemudviklingens analyse- og designarbejde.

4.2.4 Begrænsninger ved testing

Selvom usability testing udgør en central metode inden for usability engineering, er metoden forbundet med en række begrænsninger, som har betydning for, hvordan testresultater kan anvendes i systemudvikling. For det første er usability testing karakteriseret ved lav reproducerbarhed, da resultaterne påvirkes af individuelle forskelle mellem brugere samt variationer i testkonteksten. Dette gør det vanskeligt at generalisere resultaterne på tværs af testsituationer. For det andet er testens validitet afhængig af, i hvilken grad testopsætningen afspejler den faktiske brugskontekst. Hertzum fremhæver, at usability i høj grad bør vurderes

i relation til organisatoriske og praktiske sammenhænge, som ofte ikke kan reproduceres i kontrollerede testmiljøer (Hertzum, 2016, s. 84-86).

Endelig er usability testing påvirket af selve testmetoden. Som tidligere nævnt kan eksempelvis think-aloud påvirke brugerens adfærd og dermed ændre det grundlag, som analysen bygger på (Hertzum et al., 2009, s. 165-166). Samlet set betyder disse begrænsninger, at usability testing producerer værdifuld, men usikker og fortolkningskrævende viden. Disse forhold udgør et centralt udgangspunkt for specialets videre analyse, hvor udfordringen med at strukturere og omsætte usability-viden til systemudvikling behandles nærmere.

4.3 Software testing og teststandarder

Dette afsnit introducerer software testing, som den forstås inden for International Software Testing Qualifications Board (ISTQB), med fokus på de testlogikker, testprincipper og kvalitetsforståelser, der præger denne tilgang. Formålet er ikke at give en udtømmende gennemgang af testprocesser, men at etablere et teoretisk grundlag for at analysere, hvordan software testing producerer verificerende viden baseret på kravspecifikationer. Afsnittet fungerer dermed som teoretisk kontrast til usability testing og forbereder analysen af spændingsfeltet mellem de to testformer.

4.3.1 Software testing som verificerende praksis

Software testing spiller en central rolle i kvalitetssikringen af digitale systemer, idet den har til formål at vurdere systemets korrekthed og reducere risikoen for fejl. Ifølge Stapp et al. (2024, s. 27) forstås software testing inden for International Software Testing Qualifications Board (ISTQB) som en række aktiviteter, der understøtter identifikation af softwarefejl og evaluering af softwareartefakter. Formålet med software testing er at verificere, at systemet lever op til de specificerede krav, samt at validere systemets funktion i relation til forretnings- og brugerkrav. En central del af software testing er, at der tages udgangspunkt i prædefinerede kravspecifikationer og forventninger til systemets adfærd. Testens formål er derfor at vurdere, hvorvidt systemet fungerer som planlagt, snarere end hvordan systemet opleves af brugeren i praksis. Software testing producerer dermed verificerbar viden, hvor kvalitet forstås som overensstemmelse mellem systemets observerede adfærd og de krav, der er fastlagt på forhånd.

4.3.2 Testniveauer og testmål

Inden for software testing arbejdes der systematisk med testniveauer, som hver har klart definerede testmål. Ifølge Stapp et al. (2024, s. 99) udgør testniveauer grupper af testaktiviteter, der organiseres og håndteres samlet som led i en struktureret testproces. De mest udbredte testniveauer er component testing, component integration testing, system testing, system integration testing og acceptance testing. Fælles for disse testniveauer er, at de alle arbejder ud fra en verificerende testlogik, hvor systemets adfærd vurderes i forhold til tidligere fastlagte krav og forventninger. Testniveauerne adskiller sig primært ved, hvilket abstraktionsniveau der testes på, men ikke ved deres grundlæggende forståelse af kvalitet.

Uanset om testen udføres på komponent-, integrations- eller systemniveau, er målet at identificere defekter og verificere, at systemet lever op til de specificerede funktionelle og non-funktionelle krav. Testniveauerne understøtter tilsammen en gradvis reduktion af risiko i systemudviklingen, idet fejl søges identificeret så tidligt som muligt, før de spreder sig til højere niveauer. Denne opdeling bidrager samtidig til at opbygge tillid til systemets korrekthed gennem gentagne verificeringer på tværs af systemets dele. Testniveauerne fungerer således som et organiserende princip, der sikrer systematisk kontrol og verificering gennem hele udviklingsprocessen (Stapp et al., 2024, s. 99-110).

4.3.3 Testprocesser, dokumentation og sporbarhed

Software testing er karakteriseret ved en høj grad af struktur, planlægning og dokumentation. Ifølge International Software Testing Qualifications Board består software testing af en række sammenhængende testaktiviteter, der tilsammen understøtter en kontrolleret og systematisk testproces (Stapp et al., 2024, s. 48). Formålet med denne struktur er at sikre, at testresultater kan anvendes som et pålideligt grundlag for beslutninger i systemudviklingen. Et centralt element i software testing er anvendelsen af en testbasis, som udgør referencepunktet for testaktiviteterne. Testbasen består typisk af kravspecifikationer, designbeskrivelser og andre artefakter, der definerer, hvordan systemet forventes at fungere. På baggrund af dette udformes testcases med forventede resultater, som gør det muligt at sammenligne observeret systemadfærd med de fastlagte forventninger (Stapp et al., 2024, s. 49-51).

Denne kravbaserede tilgang muliggør sporbarhed mellem krav, testcases og testresultater. Sporbarhed spiller en central rolle i software testing, da den gør det muligt at dokumentere, hvilke krav der er testet, hvilke defekter der er identificeret, og hvordan disse er håndteret. Sporbarhed bygger dermed på eksplicitte relationer mellem krav, testcases,

systemkomponenter og testresultater, hvilket understøtter en struktureret og relationel forståelse af systemkvalitet.

Samtidig understøtter sporbarhed gentagelighed, idet tests kan reproduceres under tilsvarende betingelser for at verificere, om identificerede fejl er blevet rettet. Den strukturerede testproces bidrager dermed til at reducere usikkerhed i udviklingsprocessen og skabe mere entydige vurderinger af systemets korrekthed. Software testing producerer i denne forstand verificerbar viden, der er velegnet til styring, dokumentation og kvalitetssikring, men som samtidig er afgrænset til det, der kan specificeres og måles i relation til fastlagte krav.

4.3.4 Usability som non-funktionel kvalitet

Dette afsnit belyser, hvordan usability er positioneret inden for software testing i henhold til International Software Testing Qualifications Board. I ISTQB's Foundation Level Syllabus skelnes der mellem forskellige testtyper, herunder functional testing og non-functional testing, som tilsammen udgør centrale elementer i en kravbaseret og verificerende testlogik (Stapp et al., 2024, s. 115). Functional testing har til formål at verificere, hvad systemet gør, ved at evaluere funktionel korrekthed, hensigtsmæssighed og fuldstændighed i forhold til specificerede krav. Denne testtype baserer sig på eksterne artefakter såsom kravspecifikationer og brugerhistorier og anvender mål som functional coverage til at vurdere, i hvilken grad systemets funktioner er testet (Stapp et al., 2024, s. 115-116).

Non-functional testing fokuserer derimod på, hvordan systemet opfører sig, og omfatter karakteristika såsom usability, performance og sikkerhed. Usability indgår her som én blandt flere non-funktionelle kvaliteter, der kan testes og vurderes i relation til fastlagte krav. Ligesom ved functional testing anvendes begreber som coverage til at måle, i hvilken grad non-funktionelle krav er dækket af testningen (Stapp et al., 2024, s. 117-119). Inden for denne testlogik behandles usability således primært som en målbar kvalitetsdimension, der kan verificeres gennem strukturerede testprocesser. Denne forståelse adskiller sig væsentligt fra usability testing, hvor fokus er på at skabe indsigt i relationer mellem brugere, handlinger, systemfunktioner og kontekster i konkrete brugssituationer.

Hvor software testing søger at verificere opfyldelsen af specificerede krav, producerer usability testing fortolkningskrævende og relationel viden, som ikke uden videre kan reduceres til målbare størrelser som coverage. Denne forskel i forståelsen af usability bidrager til det spændingsfelt mellem software testing og usability testing, som danner grundlag for specialets videre analyse.

4.3.5 Spændingsfelt mellem usability testing og software testing

Som vist i dette afsnit fremhæver ISTQB, at software testing er præget af verificerende, kravbaserede og strukturerede testlogikker, hvor kvalitet forstås som opfyldelse af specificerede funktionelle og non-funktionelle krav. Usability indgår i denne logik som én blandt flere non-funktionelle kvaliteter og behandles primært gennem målbare størrelser som coverage. Denne forståelse adskiller sig væsentligt fra usability testing, som tidligere er beskrevet som en fortolkningsafhængig og kontekstafhængig testform. Forskellen mellem de to tilgange peger på et strukturelt spændingsfelt i systemudvikling, hvor usability-viden ikke umiddelbart lader sig integrere i etablerede testprocesser.

Samtidig synliggør denne forskel behovet for analytiske modeller, der kan strukturere usability-viden på en måde, som gør den mere anvendelig i systemudviklingens eksisterende beslutnings- og testprocesser. Dette spændingsfelt danner udgangspunkt for den efterfølgende analyse.

4.4 Objektorienteret analyse og design som strukturel tænkning

Dette afsnit introducerer objektorienteret analyse og design som en strukturel og analytisk ramme, der kan understøtte oversættelsen af usability-indsigter til strukturelle beslutninger i systemudvikling. Formålet er ikke at behandle objektorienteret analyse og design som en implementerings- eller programmeringsteknik, men som en form for strukturel tænkning, der muliggør organisering af komplekse problemområder gennem begreber som objekter, ansvar og relationer (Mathiassen et al., 2018). Afsnittet etablerer dermed det teoretiske grundlag for den efterfølgende analyse af, hvordan usability feedback kan fortolkes, struktureres og indgå i systemudviklingens analyse- og designarbejde.

4.4.1 Behovet for strukturel tænkning i systemudvikling

Som vist i de foregående afsnit eksisterer der et spændingsfelt mellem usability testing og software testing, hvor værdifuld brugerindsigt ofte er vanskelig at omsætte til strukturerede udviklingsbeslutninger i systemudviklingen. Usability testing bidrager med kontekstafhængig, relationel og fortolkningskrævende viden om brugernes handlinger, oplevelser og interaktioner, mens software testing bygger på kravbaserede, strukturerede og målbare testprocesser. Udfordringen er således ikke mangel på viden, men mangel på en struktur, der kan forbinde disse vidensformer. Denne udfordring peger på et behov for en strukturel tilgang, der kan håndtere kompleksitet, skabe overblik og understøtte systematisk organisering af ansvar og relationer i systemudviklingen.

Objektorienteret analyse og design kan i denne sammenhæng forstås som et principbaseret rammeværk, der tilbyder netop en sådan strukturel tænkning. Mathiassen et al. (2018) fremhæver, at objektorientering ikke alene bør forstås som en teknisk programmeringsmetode, men som en analytisk tilgang til at strukturere komplekse problemområder i systemudvikling. Et centralt formål med objektorienteret analyse og design er at reducere usikkerhed og skabe overblik forud for implementering ved at tydeliggøre systemets kontekst, anvendelsesområde og centrale krav (Mathiassen et al., 2018, s. 3).

Analyse og design forstås her som tæt forbundne aktiviteter, hvor analysen fokuserer på problemområdet og dets kontekst, mens designarbejdet etablerer en struktur, der muliggør videre systemudvikling. Objektorienteret tænkning bidrager dermed til at håndtere kompleksitet og skabe en fælles analytisk forståelsesramme, der kan understøtte oversættelsen af brugerindsigt til strukturelle beslutninger i systemudviklingen.

4.4.2 Objektorienteret analyse som analytisk tilgang

Objektorienteret analyse kan bruges som en tilgang til at strukturere og fortolke komplekse systemer gennem begreber som objekter, ansvar og relationer. I denne sammenhæng anvendes objekter ikke primært som tekniske konstruktioner, men som analytiske enheder, der repræsenterer fænomener i systemets kontekst og problemområde. Ifølge Mathiassen et al. (2018) udgør objektet den grundlæggende byggesten i objektorienteret analyse og fungerer som en abstraktion af centrale fænomener, der beskrives gennem identitet, tilstand og adfærd (Mathiassen et al., 2018, s. 4).

En central pointe hos Mathiassen et al. (2018) er skellet mellem analyseobjekter og designobjekter. Analyseobjekter anvendes til at beskrive fænomener uden for systemet, såsom brugere, arbejdsopgaver og organisatoriske forhold, mens designobjekter repræsenterer elementer inden for systemet, der kan implementeres teknisk (Mathiassen et al., 2018, s. 4-5). Denne skelnen understøtter en strukturel tilgang, hvor fokus i analysen rettes mod forståelse af brugernes praksis og systemets kontekst, før der træffes beslutninger om tekniske løsninger. Objektorienteret analyse er tæt knyttet til modellering af systemets kontekst gennem begreber som problem domain og application domain. Ved at gøre disse domæner eksplicitte skabes et struktureret grundlag for videre analyse og design (Mathiassen et al., 2018, s. 6-7).

For specialets analyse betyder dette, at usability feedback ikke betragtes som isolerede observationer, men som relationel viden, der kan forbindes til konkrete objekter, handlinger, ansvar og relationer i systemets problemområde.

4.4.3 Objekter, ansvar, relationer

I objektorienteret analyse og design udgør objekter den grundlæggende strukturelle enhed, der anvendes til at organisere forståelsen af et systems problemområde. Et objekt defineres som en entitet med identitet, tilstand og adfærd og fungerer som en abstraktion af fænomener i systemets kontekst, såsom brugere, data eller organisatoriske elementer (Mathiassen et al., 2018, s. 4). Gennem objekter bliver det muligt at strukturere komplekse problemstillinger på en måde, der afspejler brugernes opfattelse af virkeligheden frem for udelukkende tekniske hensyn. Et centralt aspekt af objektorienteret tænkning er, at ansvar knyttes til objekter. I stedet for at beskrive systemets funktionalitet som adskilte procedurer fordeles ansvar for handlinger og tilstande på objekter, som hver især varetager specifikke opgaver. Denne ansvarstildeling bidrager til både materiel og mental sammenhæng i systemets struktur og understøtter udvikleres forståelse af systemet (Mathiassen et al., 2018, s. 6).

Relationer mellem objekter udgør et tredje centralt element i den objektorienterede struktur. Relationer beskriver, hvordan objekter interagerer og er afhængige af hinanden, og gør det muligt at modellere systemets kompleksitet på en overskuelig måde. Ved at eksplicitere relationer bliver det lettere at identificere konsekvenser af ændringer i systemet, da ændringer i ét objekt ofte har implikationer for relaterede objekter (Mathiassen et al., 2018, s. 6-7). Denne strukturelle organisering har samtidig ligheder med ontologiske tilgange, hvor forståelsen af et domæne opbygges gennem eksplicitte begreber og relationer mellem disse. Objekter, ansvar og relationer kan dermed forstås som analytiske kategorier, der gør det muligt at organisere usability-viden på en mere systematisk måde.

I relation til usability testing er denne strukturelle tilgang særlig relevant, da brugerindsigter ofte kan omsættes til ændringer i objekters ansvar eller relationer frem for til isolerede brugergrænsefladejusteringer. Objektorienteret tænkning tilbyder dermed et analytisk grundlag for at integrere kvalitativ usability feedback i det videre analyse- og designarbejde.

4.4.4 Fra problemforståelse til strukturel model

Udgangspunktet for objektorienteret analyse er en grundlæggende forståelse af den situation eller problemstilling, som systemet skal adressere. Ifølge Mathiassen et al. (2018) starter systemudvikling ikke med entydige krav eller færdige løsninger, men med en fortolkning af brugernes situation, deres arbejdspraksis og de problemer, der opleves i den konkrete kontekst (Mathiassen et al., 2018, s. 25). I objektorienteret analyse undgår man at reducere kompleksiteten for tidligt og vælger i stedet at strukturere den ved hjælp af

analytiske modeller. Gennem en systemdefinition, hvor der udarbejdes en kort beskrivelse af computersystemet i et neutralt sprog, identificeres centrale elementer i problemområdet, herunder objekter, relationer og ansvarsområder. Dette har til formål at skabe et stabilt fundament for videre analyse og design (Mathiassen et al., 2018, s. 24).

Overgangen fra problemforståelse til strukturel model sker, når de kvalitative og kontekstafhængige indsigter oversættes til objektorienterede begreber. Objekter anvendes her som abstraktioner af centrale fænomener i systemets problem domain, mens relationer og ansvar viser, hvordan disse fænomener hænger sammen i systemets kontekst (Mathiassen et al., 2018, s. 4-7). Den strukturelle model fungerer derfor som et mellemled mellem forståelsen af brugernes situation og den efterfølgende systemudvikling. Det er dog vigtigt at understrege, at modellen ikke udgør en objektiv eller endelig repræsentation af virkeligheden. Mathiassen et al. (2018) fremhæver, at der kan eksistere mere end én systemdefinition af den samme situation, baseret på forskellige synsvinkler (Mathiassen et al., 2018, s. 38). Modellen skal derfor forstås som en analytisk fortolkning, der gør det muligt at arbejde videre med problemstillingen på en systematisk måde.

I relation til usability testing betyder dette, at usability-indsigter ikke anvendes direkte som designkrav, men indgår som input til den strukturelle modellering. Ved at relatere usability feedback til objekter, ansvar og relationer bliver det muligt at forstå brugeroplevelser som strukturelle udfordringer i systemets opbygning frem for som isolerede interfaceproblemer. Objektorienteret tænkning understøtter dermed en analytisk oversættelse fra problemforståelse til strukturel model, som kan danne grundlag for det videre analyse- og designarbejde.

4.4.5 Objektorientering som bro mellem usability og systemudvikling

Som tidligere beskrevet anvendes usability testing som en fortolkningsafhængig og kontekstafhængig testform, mens software testing typisk følger en struktureret, kravbaseret og verificerende tilgang. Disse to vidensformer har ofte vanskeligt ved at mødes i praksis, idet de producerer forskellige typer viden med forskellige formål. Objektorienteret analyse og design kan i denne sammenhæng forstås som et muligt bindeled mellem de to tilgange. Selvom objektorienteret tænkning traditionelt forbindes med implementeringsmæssige strukturer, anvendes den her som en form for strukturel tænkning. Begreber som objekter, ansvar og relationer overføres dermed til usability testing som et analytisk middel til at strukturere usability-indsigter.

Denne strukturering kan samtidig forstås i forlængelse af ontologiske og relationelle perspektiver på viden, hvor komplekse problemområder organiseres gennem eksplicitte begreber og relationer. Objektorientering bidrager dermed ikke blot med struktur, men med et analytisk sprog, der gør det muligt at forbinde usability-viden med systemudviklingens eksisterende modeller og beslutningsprocesser. Samlet set muliggør objektorienteret analyse og design en strukturel sammenkobling af usability testing og systemudvikling uden at reducere brugerindsigter til isolerede krav eller tekniske detaljer. Ved at anvende objektorienterede begreber bliver det muligt at organisere og fortolke usability-relaterede problemstillinger på en måde, der kan indgå i systemudviklingens analyse- og designarbejde.

Objektorientering fungerer således som et fælles analytisk sprog, der kan forbinde kontekstafhængig og relationel usability-viden med mere strukturerede udviklingsprocesser. Dette danner grundlaget for specialets videre analyse, hvor det undersøges, hvordan usability feedback konkret kan omsættes til strukturelle beslutninger i systemudviklingen.

4.5 Ontologi og strukturering af viden

I dette speciale forstås usability feedback som en kontekstafhængig og fortolkningsafhængig vidensform, der ikke direkte kan omsættes til strukturerede systemkrav. Dette peger på behovet for en begrebsramme, der kan understøtte strukturering og fortolkning af usability-viden. Ontologi introduceres her som en måde at organisere viden gennem begreber og relationer mellem centrale elementer i systemets problemområde. Objektorienteret analyse kan i denne sammenhæng forstås som en ontologisk inspireret tilgang, hvor problemområder struktureres gennem begreber som objekter, ansvar og relationer. Målet er ikke at reducere usability feedback til objektive fakta, men at understøtte en mere eksplicit og systematisk fortolkning af usability-viden som grundlag for analyse- og designarbejde.

4.5.1 Ontologi som begrebsramme

Ontologi kan overordnet forstås som en måde at strukturere og repræsentere viden om et domæne gennem begreber og relationer mellem disse. Inden for computer science beskriver Guarino et al. (2009) ontologi som en “explicit specification of a conceptualization”, hvor centrale begreber og relationer ekspliciteres for at skabe en systematisk forståelse af et område. Ontologier fungerer dermed som strukturerede repræsentationer af et domæne, hvor relevante elementer og deres relationer gøres eksplicitte (Guarino et al., 2009; Jakobsen & Graf, 2022).

I relation til usability testing er denne forståelse relevant, fordi usability feedback sjældent fremstår som entydig eller isoleret viden. Feedback opstår i konkrete brugssituationer og påvirkes både af brugerens oplevelse og evaluatorens fortolkning. Ud fra et ontologisk perspektiv kan usability feedback derfor forstås som relationel viden om sammenhænge mellem eksempelvis bruger, handling, funktion, problem og kontekst. Ontologisk strukturering reducerer dermed ikke behovet for fortolkning, men understøtter en mere eksplicit og systematisk analyse af usability feedback.

Ved at anlægge en ontologisk tilgang bliver det muligt at organisere usability feedback som relationer mellem centrale begreber i systemets problemområde. Dette kan understøtte identifikation af mønstre i brugernes oplevelser og analysere, hvordan usability-problemer relaterer sig til systemets opbygning og anvendelse.

4.5.2 Knowledge Graphs

Knowledge graphs kan bruges som relationelle strukturer, hvor viden organiseres gennem forbindelser mellem entiteter og relationer i et grafbaseret format (Hogan et al., 2021). Denne relationelle organisering af viden minder om Jakobsen og Grafs (2022) arbejde med ontologiske community graphs, hvor relationer mellem begreber og fortolkninger anvendes til at strukturere komplekse vidensdomæner. Information organiseres typisk som relationelle tripler bestående af subjekt, relation og objekt, hvilket gør det muligt at repræsentere viden som netværk af sammenhængende relationer frem for isolerede informationselementer (Li et al., 2024). Knowledge graphs er tæt knyttet til ontologier, idet ontologien definerer betydningen af de begreber og relationer, der indgår i grafen. Knowledge graphs kan dermed forstås som grafbaserede repræsentationer af relationer mellem begreber baseret på ontologiske strukturer.

I relation til dette speciale er knowledge graphs relevante, fordi de tilbyder en måde at strukturere usability feedback som relationel viden. I stedet for at behandle usability-observationer som isolerede problemer bliver det muligt at organisere dem som forbindelser mellem eksempelvis brugere, handlinger, funktioner, problemer og kontekster. Dette kan understøtte identifikation af mønstre og sammenhænge på tværs af usability feedback og dermed bidrage til en mere struktureret forståelse af systemets problemområde.

4.5.3 Analytisk strukturering af usability feedback

På baggrund af de foregående afsnit kan der identificeres et behov for en mere eksplicit analytisk tilgang til arbejdet med usability feedback i systemudvikling. Usability feedback er karakteriseret ved at være kontekstafhængig, kvalitativ og fortolkningskrævende, hvilket gør det vanskeligt at omsætte den direkte til strukturerede systembeslutninger. Samtidig viser ontologiske og relationelle perspektiver, at viden kan organiseres systematisk gennem begreber og relationer. På denne baggrund kan usability feedback analytisk forstås gennem tre sammenhængende perspektiver: et ontologisk perspektiv, der fokuserer på begreber og deres betydning, et relationelt perspektiv gennem knowledge graphs, hvor forbindelser mellem elementer synliggøres, samt et objektorienteret perspektiv, hvor usability feedback relateres til objekter, ansvar og relationer i systemets struktur. Tilsammen udgør disse perspektiver en analytisk ramme for strukturering og fortolkning af usability feedback i relation til systemudvikling.

4.5.3.1 Ontologisk identifikation af begreber og relationer

Det første perspektiv består i en ontologisk strukturering af usability feedback gennem identifikation af centrale begreber og relationer i systemets problemområde. Usability feedback kan fortolkes som relationer mellem elementer såsom bruger, handling, systemfunktion, problem og kontekst frem for som enkeltstående observationer. Et usability-problem opstår eksempelvis i relationen mellem en brugerhandling og en systemfunktion i en bestemt kontekst. Den ontologiske strukturering gør det muligt at tydeliggøre, hvilke begreber og relationer der er relevante for analysen, samt hvordan disse forbindes i systemets problemområde (Guarino et al., 2009). Ontologisk strukturering reducerer dermed ikke behovet for fortolkning, men understøtter en mere eksplicit og systematisk analyse af usability feedback.

4.5.3.2 Relationel strukturering gennem knowledge graphs

Det andet perspektiv består i en relationel strukturering af usability feedback gennem knowledge graphs. Her organiseres usability feedback som forbindelser mellem centrale elementer og handlinger i et netværk af relationer. Feedback kan eksempelvis repræsenteres gennem relationelle tripler såsom:

Bruger → udfører → handling
Handling → interagerer med → funktion
Funktion → relaterer til → problem

Denne strukturering gør det muligt at analysere usability feedback som sammenhængende relationer frem for som enkeltstående observationer. Som beskrevet af Hogan et al. (2021) understøtter knowledge graphs analyse af komplekse sammenhænge og mønstre på tværs af relationer mellem elementer i et domæne.

Knowledge graph-strukturen fungerer dermed som et analytisk perspektiv på, hvordan usability-viden kan organiseres og forbindes gennem relationer mellem systemets centrale elementer. Lignende relationelle strukturer beskrives af Jakobsen og Graf (2022), hvor ontologiske community graphs anvendes til at organisere fortolkningsrelationer mellem begreber og vidensobjekter.

4.5.3.3 Objektorienteret fortolkning af usability feedback

Det tredje perspektiv består i en objektorienteret fortolkning af den strukturerede usability feedback. Her fortolkes relationerne mellem bruger, handlinger, funktioner og problemer gennem objektorienterede begreber såsom objekter, ansvar og relationer. Dette gør det muligt at relatere usability feedback til systemets strukturelle opbygning frem for alene til brugergrænsefladen. Som beskrevet i afsnit 4.4 tilbyder objektorienteret analyse et strukturelt sprog til at forstå, hvordan usability-problemer relaterer sig til systemets strukturelle organisering (Mathiassen et al., 2018).

I denne sammenhæng fungerer knowledge graph-strukturen som et analytisk mellemled mellem usability-observationer og systemets strukturelle model. Fokus flyttes dermed fra enkelte brugeroplevelser til de relationer og strukturelle forhold i systemet, som bidrager til usability-problemerne.

4.5.3.4 Sammenkobling af de analytiske perspektiver

Samlet set beskriver de tre perspektiver en sammenkobling mellem ontologi, knowledge graphs og objektorienteret analyse. Usability feedback forstås ontologisk som relationer mellem begreber, struktureres relationelt gennem knowledge graphs og fortolkes objektorienteret som strukturelle problemstillinger i systemet. De tre perspektiver skal dog ikke forstås som adskilte trin eller som en fast metode, men som analytiske perspektiver, der understøtter fortolkning og strukturering af usability feedback. Tilgangen fungerer dermed som en begrebslig og analytisk ramme, der kan understøtte refleksion over, hvordan usability-viden kan relateres til systemudviklingens analyse- og designarbejde.

Denne sammenkobling bidrager til at besvare specialets problemformulering ved at tilbyde en tilgang til, hvordan usability feedback kan analyseres og struktureres, samtidig med at objektorienteret analyse fungerer som bindeled mellem usability-viden og systemudvikling.

5. Analyse

5.1 Karakteren af usability feedback

Dette afsnit analyserer karakteren af den viden, der produceres gennem usability testing, og diskuterer, hvorfor denne vidensform er metodisk og analytisk udfordrende i systemudvikling. Fokus er ikke på konkrete testmetoder, men på den type viden usability feedback repræsenterer, samt hvilke implikationer dette har for dens anvendelse i udviklingsprocesser. I dette speciale forstås usability feedback som fortolkningsafhængig og kontekstuel viden om samspillet mellem bruger og system.

5.1.1 Usability feedback som kontekstafhængig viden

Usability testing producerer kvalitativ viden baseret på observation af, hvordan brugere udfører opgaver, fortolker systemets funktioner og reagerer på dets struktur i praksis. I modsætning til software testing vurderes det ikke, om systemets funktioner teknisk fungerer som specificeret, men hvordan de opleves og anvendes af brugerne. Et system kan eksempelvis være teknisk korrekt og samtidig vanskeligt at anvende, hvis brugerne har problemer med at finde, forstå eller anvende funktionerne effektivt.

Usability feedback kan pege på problemer såsom manglende synlighed, uhensigtsmæssig placering af funktioner eller uklare sammenhænge mellem handling og feedback. Observationerne udgør dog ikke objektive eller endelige vurderinger. Som Hertzum og Clemmensen (2012) påpeger, afhænger vurderingen af usability af, hvilke aspekter af systemets anvendelse der fremhæves som relevante i den konkrete kontekst.

Feedbacken er samtidig kontekstafhængig. Et problem kan vurderes forskelligt afhængigt af systemets formål, målgruppe og anvendelsessituation. Et learnability-problem kan eksempelvis være mindre kritisk i et ekspertsystem end i et system målrettet uerfarne brugere. Usability feedback må derfor forstås som relationel viden om samspillet mellem bruger, system og kontekst.

Usability feedback kan desuden udvindes gennem forskellige kilder under testen. Think-aloud-metoden giver indsigt i brugerens overvejelser, mens observationer kan afdække problemer, som brugeren ikke selv er bevidst om. Samtidig påvirkes feedbacken af selve testmetoden, idet think-aloud ændrer brugerens refleksionsproces og dermed også den type viden, der produceres under testen (Hertzum et al., 2009).

5.1.2 Fortolkningsbehov og usikkerhed

Den viden, der produceres gennem usability testing, kan ikke anvendes direkte uden fortolkning. Observationerne skal sammenholdes og analyseres på tværs af testbrugere, før de kan omsættes til systemudvikling. Hvor software testing resulterer i entydige afgørelser om, hvorvidt en funktion virker eller ej, producerer usability testing beskrivelser af problemområder, som kræver analytisk vurdering.

Usability feedback kan eksempelvis vise, at flere brugere bruger lang tid på at udføre en opgave, uden at det i sig selv forklarer, hvad der bør ændres i systemet. Udviklere må derfor analysere, om problemet skyldes manglende information, uklare relationer mellem systemets dele eller uhensigtsmæssig struktur. Feedbacken kan dermed forstås som relationel viden om samspillet mellem bruger og system frem for som isolerede problemer.

Fortolkningen indebærer samtidig usikkerhed, da forskellige brugere og evaluatorer kan identificere og prioritere forskellige usability-problemer i samme testsituation (Hertzum & Clemmensen, 2012). Usability testing kan derfor ikke verificere systemets korrekthed, men undersøger hvorvidt systemet opleves som anvendeligt og forståeligt i praksis.

Manglen på en indbygget struktur til prioritering og beslutningstagning betyder samtidig, at usability feedback ikke i sig selv angiver, hvilke problemer der bør løses, eller hvordan de bør håndteres. Uden en supplerende analytisk ramme risikerer usability feedback derfor at blive reduceret til mindre brugergrænsefladejusteringer frem for mere strukturelle overvejelser.

5.1.3 Delkonklusion: usability feedback som analytisk udfordring

Samlet set fremstår usability feedback som en værdifuld, men metodisk udfordrende vidensform i systemudvikling. Usability testing bidrager med indsigt i brugernes faktiske interaktion med systemet og afdækker problemområder, der ikke nødvendigvis identificeres gennem kravbaserede eller tekniske testmetoder. Samtidig er denne viden præget af variation, kontekstafhængighed og fortolkningsbehov, hvilket gør den vanskelig at omsætte direkte til beslutninger i systemudviklingen. Hvor software testing producerer klare afgørelser om systemets korrekthed, producerer usability testing beskrivelser af problemområder, som peger på, hvor problemer opstår, men ikke hvordan de bør håndteres.

Dette skaber et analytisk spændingsfelt, hvor usability feedback er central for forståelsen af systemets anvendelighed i praksis, men samtidig mangler en ramme, der kan understøtte organiseringen af relationer mellem bruger, system og kontekst. Dette peger på behovet for

et analytisk mellemled, hvor usability feedback kan struktureres som relationel viden gennem begreber og relationer, før den omsættes til systembeslutninger. I det følgende afsnit undersøges det derfor, hvordan objektorienteret analyse og design kan fungere som en strukturel ramme, hvor begreber som objekter, ansvar og relationer anvendes til at forbinde brugerindsigter med systemets overordnede struktur.

5.2 Spændingsfeltet mellem usability testing og software testing

Dette afsnit analyserer det strukturelle spændingsfelt mellem usability testing og software testing ved at sammenholde deres respektive testlogikker, vidensformer og funktion i systemudvikling. Formålet er at vise, hvordan de to testformer producerer fundamentalt forskellige typer viden, og hvorfor denne forskel skaber udfordringer, når usability feedback skal integreres i kravbaserede og verificerende udviklingsprocesser.

5.2.1 Software testings verificerende logik

Software testing har til formål at verificere, hvorvidt et system lever op til de krav og forventninger, der er fastlagt forud for udviklingen. Ifølge International Software Testing Qualifications Board (ISTQB) defineres software testing som en række aktiviteter, der har til hensigt at opdage defekter og evaluere softwareartefakter for at sikre, at systemet opfylder specificerede krav (Stapp et al., 2024, s. 27). Testprocessen tager således udgangspunkt i kravspecifikationer, designbeskrivelser og forventede resultater og afsluttes med en vurdering af, om disse er opfyldt. Den verificerende karakter indebærer, at testresultater vurderes i relation til faste referencepunkter. Systemets observerede adfærd sammenholdes med de forventede resultater, og udfaldet klassificeres som enten bestået eller ikke bestået. Denne binære testlogik er grundlæggende for software testing og muliggør klar dokumentation, sporbarhed og gentagelighed af testresultater.

Den verificerende logik afspejles også i opdelingen af testniveauer, herunder component testing, component integration testing, system testing, system integration testing og acceptance testing. På tværs af disse niveauer er formålet at reducere risiko ved at identificere fejl og verificere systemets korrekthed i forhold til funktionelle og non-funktionelle krav (Stapp et al., 2024, s. 99-110). Den viden, software testing producerer, er dermed teknisk anvendelig og handlingsorienteret, men samtidig afgrænset. Testresultater giver klare afgørelser om, hvorvidt en funktion, komponent eller systemadfærd fungerer som forventet. Der er begrænset plads til fortolkning, idet afvigelser fra krav kan identificeres og omsættes direkte til tekniske rettelser. Til gengæld producerer software testing ikke viden om, hvordan systemet opleves eller anvendes af brugere i praksis.

5.2.2 Usability testings diagnostiske logik

Modsat software testing er usability testing karakteriseret ved en diagnostisk testlogik. Her er formålet ikke at verificere korrekthed, men at identificere, hvor og hvordan problemer opstår i brugerens interaktion med systemet. Usability testing afdækker problemområder i konkrete brugssituationer, men angiver ikke i sig selv, hvordan disse problemer bør løses. Som Hertzum og Clemmensen (2012, s. 26-27) fremhæver, er usability findings ikke stabile eller entydige observationer, men afhænger af evaluatorens perspektiv og den konkrete testsituation. En central forskel mellem de to testformer er forståelsen af, hvad der udgør et problem. I software testing refererer en fejl til en afvigelse fra specificerede krav eller forventet systemadfærd. I usability testing anvendes problemer bredere og kan omfatte misforståelser, manglende overblik, frustration eller uhensigtsmæssige interaktionsforløb. Et system kan således være teknisk korrekt og samtidig skabe betydelige problemer i brugerens anvendelse af systemet.

Usability testing producerer ikke data, der kan verificeres gennem faste referencepunkter. Usability-problemer varierer mellem brugere og afhænger af kontekst, erfaring og opgavesituation. Resultaterne kan derfor ikke reduceres til binære afgørelser om korrekthed, men kræver analyse af observationer, handlinger og brugerudsagn. I stedet producerer usability testing relationel viden om samspillet mellem bruger, handling, systemfunktion og kontekst. Denne testlogik indebærer, at usability feedback ikke automatisk kan omsættes til konkrete systembeslutninger. Observationer og usability findings må først fortolkes i relation til systemets funktioner, brugssituationer og problemområder. Usability testing resulterer derfor ikke i entydige afgørelser, men i indsigter, som systemudviklere må analysere og anvende i udviklingsprocessen.

5.2.3 Analytisk kontrast mellem testlogikkerne

Forskellene mellem software testing og usability testing kan ikke reduceres til metodiske variationer, men afspejler dybere forskelle i formål, vidensgrundlag og forståelsen af, hvad der udgør et problem i et system. Software testing søger at reducere variation gennem kontrol, reproducerbarhed og stabile testbetingelser. Variation i testresultater opfattes som støj, der kan sløre vurderingen af systemets korrekthed (Stapp et al., 2024, s. 307-308). Testens kvalitet afhænger derfor af graden af kontrol, da dette muliggør gentagelse, sammenligning og verificering af resultater. Usability testing arbejder med en anden logik. Her accepteres og forventes variation mellem testbrugere, da forskelle i erfaring, strategier og adfærd udgør centrale datakilder.

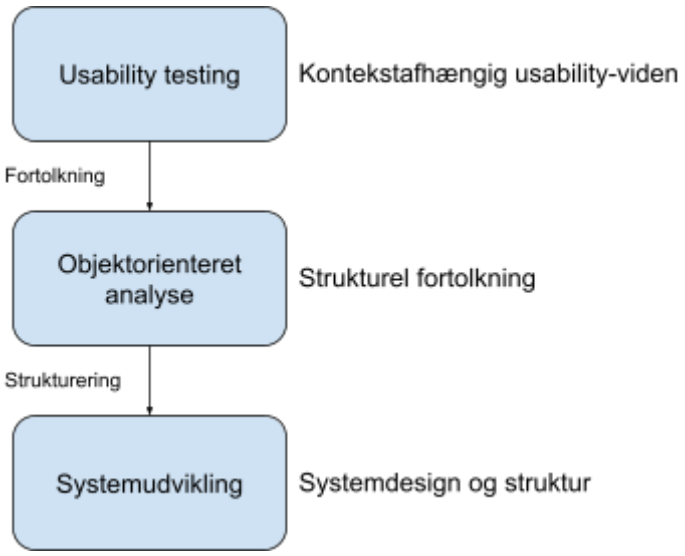
Ét usability-problem kan manifestere sig forskelligt på tværs af brugere, men disse variationer betragtes som forskellige indsigter i samme underliggende problem. Testens styrke ligger derfor ikke i kontrol, men i analytisk åbenhed over for kontekst og fortolkning. Disse forskelle fører til forskellige typer testresultater. Software testing resulterer i testcases, fejlrapporter og dokumenterede afvigelser fra krav, som kan omsættes direkte til tekniske rettelser. Usability testing resulterer derimod i observationer og beskrivelser af brugeradfærd, som først må analyseres, sammenlignes og fortolkes, før de kan anvendes i systemudviklingen. Denne forskel peger på behovet for en struktur, der kan organisere usability feedback som relationer mellem centrale elementer i systemets problemområde frem for som isolerede observationer.

5.2.4 Konsekvenser i praksis og behovet for et mellemled

I praksis får software testing ofte større beslutningskraft i systemudvikling, da testresultaterne er lette at dokumentere, reproducere og anvende som grundlag for projektstyring og milepælsbeslutninger. Usability testing producerer derimod viden, der kræver fortolkning og analytisk arbejde, hvilket kan gøre det vanskeligere at integrere i tids- og processtyrede udviklingsforløb. Som konsekvens anvendes usability testing ofte sent i udviklingsprocessen og reduceres til overfladiske justeringer af brugergrænseflader, uden at underliggende strukturelle problemer nødvendigvis adresseres.

Spændingsfeltet mellem usability testing og software testing opstår derfor ikke blot som et metodisk problem, men som et strukturelt problem i måden, hvorpå viden anvendes og prioriteres i systemudvikling. Spændingsfeltet handler dermed også om forskellige måder at strukturere og forstå viden på i udviklingsprocessen. Dette peger på behovet for et analytisk mellemled, hvor usability feedback kan struktureres gennem begreber og relationer, før den fortolkes i relation til systemets struktur og omsættes til systembeslutninger.

Som illustreret i figur 1 kan objektorienteret analyse forstås som et analytisk mellemled, hvor usability feedback analyseres og struktureres gennem objekter, ansvar og relationer. Denne analytiske strukturering gør det muligt at relatere brugerindsigter til systemets problemdomæne og dermed understøtte systemudviklingens analyse- og beslutningsprocesser.



Figur 1: Objektorienteret analyse som analytisk mellemled mellem usability feedback og systemudvikling.

5.3 Objektorienteret analyse som strukturel oversættelse

De foregående afsnit har vist, at usability testing og software testing producerer forskellige typer viden, som er vanskelige at integrere direkte i systemudviklingens beslutningsprocesser. Dette afsnit introducerer objektorienteret analyse som et analytisk mellemled, der kan understøtte oversættelsen af usability feedback til en mere strukturel forståelse af systemets problemområder. Objektorienteret analyse anvendes her som en strukturerende og fortolkende ramme, hvor usability feedback analyseres gennem objekter, ansvar og relationer. Afsnittet undersøger først objekter som analytiske enheder, dernæst ansvar som fortolkningsramme og afslutningsvis relationer som struktur for kompleksitet.

5.3.1 Objekter som analytiske enheder

Usability testing kan resultere i feedback om forvirring, frustration eller vanskeligheder ved at gennemføre opgaver. Denne feedback beskriver oplevelser og observationer, men angiver ikke i sig selv, hvor i systemets struktur problemerne opstår. Uden en sådan placering risikerer usability feedback at blive reduceret til isolerede justeringer af brugergrænsefladen frem for mere grundlæggende strukturelle overvejelser.

Objektorienteret analyse gør det muligt at flytte fokus fra brugerens oplevelse til de elementer i systemets struktur, der er involveret i problemet. I objektorienteret analyse

forstås et objekt som en abstraktion af et fænomen i problemområdet, karakteriseret ved identitet, tilstand og adfærd (Mathiassen et al., 2018, s. 51-52).

Overført til usability testing betyder dette, at analysen retter sig mod de fænomener i systemet, som brugeren interagerer med. I et ontologisk perspektiv kan objekter forstås som centrale begrebslige enheder i systemets problemområde, hvor usability feedback organiseres gennem relationer mellem bruger, handling og systemfunktion (Guarino et al., 2009, s. 2-3).

Hvis flere brugere eksempelvis bliver forvirrede over en bestemt del af systemet, kan analysen rettes mod de objekter, der har ansvar for at formidle information eller understøtte brugerens handlinger. Problemet forankres dermed analytisk i systemets struktur frem for alene i brugeroplevelsen.

5.3.2 Ansvar som fortolkningsramme

I objektorienteret analyse beskriver ansvar, hvilke opgaver og handlinger et objekt varetager, samt hvordan det reagerer på input (Mathiassen et al., 2018, s. 41). Ansvar handler dermed om, hvordan systemet understøtter brugerens handlinger gennem information, feedback og respons. I relation til usability testing fungerer ansvar som et analytisk greb, der muliggør en strukturel fortolkning af brugeroplevelser. Når en bruger eksempelvis udtrykker forvirring eller frustration under en usability test, kan dette fortolkes som et tegn på, at systemet ikke har opfyldt sit ansvar i den givne situation. Analysen kan dermed bevæge sig fra udsagn som “brugeren blev forvirret” til formuleringer som “systemet opfyldte ikke sit ansvar for at formidle relevant information”.

Denne forskydning flytter fokus væk fra individuelle brugeroplevelser og hen imod systemets strukturelle opbygning. Samtidig bevares usability feedbackens kontekstuelle karakter, idet ansvar ikke angiver konkrete løsninger, men peger på, hvilke aspekter af systemets adfærd der bidrager til problemerne. I relation til specialets ontologiske perspektiv kan ansvar forstås som relationer mellem objekter og handlinger i systemets problemområde. Ansvar fungerer dermed som en strukturerende mekanisme, der gør det muligt at forbinde usability feedback med systemets funktionelle og logiske opbygning. Ansvar fungerer således som et analytisk mellemled mellem oplevelsesbaseret feedback og systemudvikling og åbner for analyse af samspillet mellem flere objekter.

5.3.3 Relationer som struktur for kompleksitet

Usability-problemer kan sjældent forklares ud fra ét enkelt objekt og dets ansvar. I mange tilfælde opstår problemer i samspillet mellem systemets objekter. Et system kan eksempelvis indeholde de nødvendige informationer og give korrekt feedback, men stadig skabe forvirring, hvis informationerne ikke præsenteres i en rækkefølge, der understøtter brugerens handlinger. I objektorienteret analyse beskrives relationer som forbindelser mellem objekter, der viser, hvordan objekter påvirker hinanden, og hvordan ændringer i ét objekt kan få konsekvenser for andre (Mathiassen et al., 2018, s. 76-77). Relationer gør det muligt at analysere systemet som en helhed frem for som isolerede dele.

Overført til usability testing betyder dette, at problemer ikke nødvendigvis skyldes manglende funktionalitet i et enkelt objekt, men relationerne mellem information, handling og feedback. Usability feedback kan dermed forstås som et netværk af relationer mellem bruger, handling, funktion og kontekst. Denne forståelse ligger tæt op ad knowledge graphs, hvor viden organiseres gennem relationer mellem centrale begreber i et domæne (Hogan et al., 2021, s. 4). Ved at inddrage relationer som analytisk begreb kan usability feedback fortolkes som strukturelle udfordringer på systemniveau frem for som isolerede brugergrænsefladeproblemer.

5.4 Fra indsigt til strukturel model

Dette afsnit fokuserer på oversættelsen fra brugerindsigt til strukturel model. Formålet er at undersøge, hvordan usability feedback gennem analytisk fortolkning kan omsættes til strukturelle repræsentationer af systemets problemområder. Usability feedback forstås her som relationel viden om sammenhænge mellem bruger og system. Oversættelsen fra indsigt til model handler derfor om at organisere og fortolke disse relationer på en måde, der gør dem anvendelige i analyse- og designarbejde.

5.4.1 Feedback som problemforståelse

Usability feedback tager udgangspunkt i brugernes oplevelser og observationer af, hvordan et system anvendes i praksis. Feedbacken beskriver, hvad der opleves som vanskeligt, og hvor interaktionen bryder sammen, men forklarer ikke i sig selv, hvorfor problemerne opstår i systemets struktur. Observationerne peger på, hvor problemer manifesterer sig i interaktionen, men synliggør ikke nødvendigvis, hvordan systemets objekter, ansvar og relationer bidrager til disse problemer. I et relationelt perspektiv kan usability feedback forstås som beskrivelser af forbindelser mellem brugerhandlinger, systemfunktioner og kontekstuelle forhold.

Et usability-problem kan eksempelvis komme til udtryk som forvirring eller frustration, uden at det er klart, om årsagen ligger i informationsstrukturen, i fordelingen af ansvar mellem systemets elementer eller i relationerne mellem flere dele af systemet. Usability feedback fungerer dermed som et symptomniveau i analysen: feedbacken synliggør, at der eksisterer et problem, men ikke hvordan det er strukturelt forankret.

Uden yderligere analytisk bearbejdning risikerer usability feedback derfor at forblive på oplevelsesniveauet og blive anvendt som grundlag for isolerede designjusteringer frem for mere grundlæggende strukturelle overvejelser.

5.4.2 Fra problemforståelse til strukturel repræsentation

Når usability testing er gennemført, kan udvikleren analysere feedbacken med henblik på at identificere mønstre på tværs af de observerede problemområder. I stedet for at betragte observationer isoleret formuleres problemerne på et mere abstrakt niveau, eksempelvis som manglende overblik, uklare handlingsforløb eller inkonsistent feedback i systemet. Det er i denne overgang, at begreberne fra objektorienteret analyse får analytisk betydning. Ved at anvende objekter, ansvar og relationer som fortolkningsramme bliver det muligt at undersøge, hvilke dele af systemets struktur der er involveret i de identificerede problemer. Analysen retter sig dermed ikke mod brugerens oplevelse alene, men mod hvordan systemets opbygning understøtter eller modarbejder brugerens handlinger.

Denne oversættelse kræver analytisk fortolkning. Usability feedback omsættes ikke direkte til løsninger eller krav, men til en strukturel repræsentation af problemet. Problemet formuleres således ikke længere som “brugeren er forvirret”, men som et problem i systemets struktur, eksempelvis uklart ansvar for informationsformidling eller problematiske relationer mellem funktionelle elementer. Eksempelvis kan en usability-observation som: “Brugeren har vanskeligt ved at finde patienthistorik”.

fortolkes objektorienteret som et problem relateret til objektet *patientjournal*, ansvar for informationsadgang samt relationerne mellem navigation, workflow og informationsstruktur. Problemet forstås dermed ikke alene som brugerforvirring, men som et strukturelt problem i organiseringen af systemets informationsadgang. Den strukturelle repræsentation fungerer dermed som et fælles analytisk grundlag for videre analyse og design.

I denne sammenhæng kan knowledge graphs forstås som en mulig måde at repræsentere relationer mellem bruger, handling, funktion og problem. Pointen er ikke den tekniske implementering, men den relationelle organisering af usability-viden, hvor problemer analyseres som forbindelser mellem flere dele af systemets struktur. På denne måde

transformeres brugerindsigt fra oplevelsesbaseret feedback til et analytisk objekt, der kan indgå i systemudviklingens videre arbejde.

5.4.3 Strukturelle modeller som analytisk arbejdsredskab

De strukturelle modeller, der udvikles gennem objektorienteret analyse, skal ikke forstås som endelige eller objektive beskrivelser af systemet. Som Mathiassen et al. (2018, s. 38) fremhæver, kan der eksistere flere systemdefinitioner af den samme problematiske situation afhængigt af perspektiv, formål og analytisk fokus. Den strukturelle model er derfor en analytisk fortolkning frem for en objektiv gengivelse af virkeligheden. Denne fortolkende karakter udgør samtidig en styrke i arbejdet med usability feedback. Da brugeroplevelser er kontekstafhængige og kan fortolkes forskelligt, giver strukturelle modeller mulighed for at undersøge og sammenligne forskellige forståelser af, hvordan problemer hænger sammen med systemets opbygning.

Modellerne fungerer som analytiske arbejdsredskaber, der understøtter refleksion, diskussion og revision af problemforståelsen. I modsætning til kravspecifikationer, der skal være entydige og verificerbare, understøtter strukturelle modeller en åben og eksplorativ analyse. De muliggør en gradvis præcisering af problemer, før der træffes beslutninger om ændringer i systemets design eller struktur. I et ontologisk perspektiv kan de strukturelle modeller samtidig forstås som eksplicitte repræsentationer af relationer mellem centrale elementer i systemets problemområde. Modellerne fungerer dermed ikke blot som beskrivelser af systemet, men som organiserede fortolkninger af usability-viden, hvor relationer mellem objekter, ansvar og handlinger synliggøres og kan analyseres systematisk.

På denne måde fungerer strukturelle modeller som et analytisk mellemled mellem usability-indsigter og systemudvikling, idet de oversætter oplevelsesbaseret feedback til en form, der kan anvendes i analyse og design uden at miste sin kontekstuelle forankring.

5.5 Analytiske cases

For at konkretisere specialets analytiske framework anvendes i dette afsnit to cases fra eksisterende usability-litteratur. Formålet er ikke at evaluere systemerne i sig selv, men at demonstrere, hvordan usability feedback kan fortolkes som strukturel og relationel viden. Den første case fokuserer på informationsarkitektur og navigation i webbaserede guideline-systemer, mens den anden fokuserer på workflows og informationsflow i et elektronisk patientjournalsystem. Begge cases analyseres gennem specialets objektorienterede, relationelle og ontologiske perspektiver.

5.5.1 Case 1 - Usability Evaluation of Published Clinical guidelines on the Web: A Case Study

Denne case tager udgangspunkt i artiklen *Usability Evaluation of Published Clinical Guidelines on the Web: A Case Study* (Khodambashi & Nytrø, 2016), der undersøger usability-relaterede udfordringer i webbaserede kliniske guideline-systemer. Evalueringen fokuserer særligt på navigation, informationsstruktur og brugerorientering i systemet. Casen er relevant, fordi usability-problemerne relaterer sig til samspillet mellem informationsstruktur, navigation og brugerens orientering på tværs af systemets forskellige dele frem for til enkelte funktioner isoleret.

Artiklen beskriver blandt andet problemer relateret til lange tekstsider, manglende navigation, inkonsistente strukturer og uklare forbindelser mellem informationsområder. Brugerne fremhæver eksempelvis udfordringer ved manglende navigation bars, omdirigering mellem forskellige websites samt vanskeligheder ved at orientere sig i store informationsmængder (Khodambashi & Nytrø, 2016, s. 143-144). Problemerne fremstår dermed ikke som tekniske fejl i systemets funktionalitet, men som problemer i relationerne mellem bruger, information, navigation og systemstruktur.

I et ontologisk og relationelt perspektiv kan usability-problemerne forstås som forbindelser mellem centrale elementer i systemets problemområde, hvor brugerens handlinger påvirkes af systemets informationsstruktur og navigationslogik. Casen danner dermed grundlag for at analysere, hvordan usability feedback kan fortolkes gennem objektorienterede begreber såsom objekter, ansvar og relationer.

5.5.1.1 Relationelle usability problemer i informationsarkitekturen

I evalueringen fremhævede brugerne problemer relateret til store mængder tekst, omfattende scrolling, navigation bars og omdirigering til eksterne hjemmesider. Khodambashi og Nytrø (2016, s. 140) beskriver eksempelvis, hvordan brugerne var:

> “Users were mostly concerned about the amount of text and scrolling, font size, no more than one navigation bar and no redirection to other websites.”

Problemerne kan umiddelbart fremstå som traditionelle usability-fejl relateret til layout, navigation og visuel organisering af information. Feedbacken peger dog bredere på problemer i relationerne mellem informationsstruktur, navigation og brugerens orientering på tværs af systemets forskellige dele. Når brugeren omdirigeres til eksterne hjemmesider med andre navigationsprincipper og informationshierarkier, opstår der brud i forståelsen af

systemets samlede struktur og sammenhæng. Casen indikerer dermed, at usability-problemerne ikke primært skyldes fejl i enkelte funktioner, men manglende sammenhæng mellem systemets informationsstrukturer og navigationslogik. Usability feedbacken beskriver derfor ikke blot problemer i brugergrænsefladen, men strukturelle problemer i relationerne mellem bruger, information og systemstruktur.

5.5.1.2 Ontologisk og relationel strukturering af usability feedback

Med udgangspunkt i den ontologiske tilgang kan usability feedback struktureres gennem identifikation af centrale begreber og relationer i systemets problemområde. I casen kan der identificeres en række centrale analytiske objekter, som indgår i brugerens interaktion med systemet:

Analytisk Objekt	**Funktion**
Bruger	Søger og anvender klinisk information
Guideline system	Organiserer og præsenterer guidelines
Navigationssystem	Understøtter orientering og adgang til information
Søgefunktion	Finder relevant information
Eksterne hjemmesider	Præsenterer relateret guideline information
Informations hierarki	Organiserer og strukturerer indhold

I stedet for at forstå usability feedback som enkeltstående observationer bliver det dermed muligt at analysere problemerne som relationer mellem objekter i systemets struktur. Problemer med orientering opstår eksempelvis i relationen mellem brugerens navigation og skiftet mellem forskellige informationsstrukturer. Tilsvarende relaterer problemer med scrolling sig ikke alene til mængden af tekst, men til hvordan information prioriteres, organiseres og forbindes i systemet.

Feedbacken kan samtidig beskrives relationelt gennem forbindelser såsom:

- Navigation → understøtter → brugerorientering
- Guideline-system → forbinder → informationsområder
- Eksterne hjemmesider → ændrer → navigationsstruktur
- Informationsstruktur → påvirker → brugerens orientering

Denne relationelle struktur synliggør, hvordan usability-problemer opstår gennem forbindelser mellem flere dele af systemets informationsarkitektur frem for i enkelte funktioner isoleret. Usability feedback kan dermed forstås som relationel viden om sammenhænge mellem bruger, information, navigation og systemstruktur.

5.5.1.3 Objektorienteret fortolkning af usability-problemerne

Objektorienteret analyse gør det muligt at fortolke usability feedback som strukturelle problemstillinger relateret til objekter, ansvar og relationer i systemet. I casen bliver det tydeligt, at flere objekter deler ansvar for brugerens orientering og informationsforståelse. Navigationssystemet understøtter brugerens bevægelser gennem systemet, mens guideline-systemet har ansvar for at præsentere information på en måde, der understøtter brugerens forståelse.

Når brugeren omdirigeres til eksterne hjemmesider med andre navigationsprincipper og informationshierarkier, fragmenteres dette ansvar på tværs af flere systemdele. Problemet opstår dermed ikke nødvendigvis, fordi enkelte funktioner fejler, men fordi relationerne mellem objekterne ikke understøtter en sammenhængende brugeroplevelse. Casen illustrerer dermed, hvordan usability feedback kan fortolkes som et strukturelt problem i relationerne mellem systemets objekter og informationsstrukturer frem for som et isoleret brugergrænsefladeproblem.

5.5.1.4 Delkonklusion

Casen viser, hvordan usability-problemer i informationstunge systemer opstår i relationerne mellem navigation, informationsstruktur og brugerorientering frem for i enkelte funktioner isoleret. Analysen understøtter dermed specialets argument om, at usability feedback kræver analytisk strukturering for at kunne omsættes til kvalificerede systembeslutninger.

5.5.2 Case 2 - Evaluating usability of a commercial electronic health record: A case study

Denne case tager udgangspunkt i artiklen *Evaluating Usability of a Commercial Electronic Health Record: A Case Study* (Edwards et al., 2008), der undersøger usability-relaterede udfordringer i et elektronisk patientjournalsystem (EHR-system). Analysen fokuserer på, hvordan sundhedsprofessionelle interagerer med systemet gennem dokumentation, informationssøgning og koordinering af kliniske aktiviteter.

Casen er relevant, fordi usability-problemerne relaterer sig til samspillet mellem arbejdsprocesser, informationsflow og systemets organisering af kliniske aktiviteter frem for til brugergrænsefladen alene. Artiklen illustrerer dermed, hvordan usability feedback i organisatoriske systemer ofte beskriver strukturelle problemstillinger på tværs af både tekniske og organisatoriske forhold.

5.5.2.1 Workflow som usability-kontekst

Et centralt aspekt i casen er, at usability ikke kan fortolkes isoleret fra de arbejdsprocesser, som systemet understøtter. Det elektroniske patientjournalsystem fungerer som en integreret del af klinikkens daglige arbejde, hvor dokumentation, informationsdeling og beslutningstagning er tæt forbundne aktiviteter. Workflowet kan dermed forstås som et samspil mellem handlinger, objekter og informationsstrømme, der understøtter klinisk praksis. Når brugerne oplever usability-problemer, handler det derfor ikke nødvendigvis om enkelte funktioner, men om hvordan systemets forskellige dele understøtter eller forstyrrer arbejdsflowet som helhed.

Klinikeren kan eksempelvis være nødt til at skifte mellem patientjournal, dokumentationsmoduler og notifikationer for at gennemføre én samlet arbejdsopgave. Problemet relaterer sig dermed ikke blot til enkelte brugergrænsefladeelementer, men til hvordan systemets struktur organiserer og forbinder arbejdsopgaver og information. Casen peger dermed på, at usability-problemer i organisatoriske systemer ofte opstår gennem relationer mellem arbejdspraksis, informationsflow og systemstruktur.

5.5.2.2 Strukturel analyse af ansvar og relationer

Med udgangspunkt i specialets analytiske framework kan følgende centrale objekter identificeres i systemets problemområde:

Analytisk objekt	**Funktion**
Bruger	Udfører kliniske arbejdsopgaver
EHR-system	Understøtter dokumentation og informationshåndtering
Patientjournal	Organiserer patientdata
Workflow	Strukturerer kliniske aktiviteter

Informationsflow	Forbinder arbejdsopgaver og information
Opgaver	Repræsenterer klinisk handling
Notifikationer	Kommunikerer ændringer og opdateringer

Usability feedbacken kan analyseres som relationer mellem objekter frem for som isolerede observationer. Problemer med informationssøgning relaterer sig eksempelvis til samspillet mellem kliniker, workflow og patientjournal, mens ineffektive arbejdsgange kan observeres som problemer i relationerne mellem arbejdsopgaver, informationsflow og systemets organisering af aktiviteter (Edwards et al., 2008). Objektorienteret analyse gør det samtidig muligt at undersøge, hvordan ansvar er fordelt mellem systemets objekter. EHR-systemet har eksempelvis ansvar for dokumentation og informationsadgang, mens workflowet strukturerer rækkefølgen af kliniske aktiviteter.

Problemer opstår, når relationerne mellem disse objekter ikke understøtter en sammenhængende arbejdsproces. Hvis information er spredt mellem flere dele af systemet, eller hvis brugeren må gennemføre unødvendige skift mellem funktioner for at løse én samlet opgave, bliver ansvaret for orientering og opgaveløsning uklart fordelt mellem systemets objekter.

5.5.2.3 Fra usability-problem til organisatorisk struktur

Casen understøtter dermed Hertzum og Clemmensens forståelse af *organizational usability*, hvor usability vurderes i relation til systemets understøttelse af organisatoriske arbejdsprocesser og praksisser (Hertzum & Clemmensen, 2012, s. 26-27). Feedbacken peger dermed på problemer i systemets organisatoriske sammenhæng frem for alene i dets brugergrænseflade. Usability-problemerne opstår ikke nødvendigvis, fordi enkelte funktioner er teknisk utilstrækkelige, men fordi relationerne mellem systemets objekter ikke understøtter brugernes arbejdspraksis på en sammenhængende måde. Problemerne kan dermed ikke forstås som isolerede fejl i brugergrænsefladen, men må analyseres i relation til workflow, informationsflow og systemets strukturelle organisering.

5.5.2.4 Delkonklusion

Casen viser, hvordan usability-problemer i organisatoriske systemer opstår i relationerne mellem workflows, informationsflow og systemansvar frem for i enkelte

brugergrænsefladeelementer. Analysen understøtter dermed specialets argument om, at usability feedback kræver analytisk strukturering for at kunne omsættes til kvalificerede systembeslutninger.

5.5.3 Tværgående analytiske pointer fra casene

På tværs af de to cases bliver det tydeligt, at usability-problemer ofte opstår i relationerne mellem systemstruktur, informationsflow og arbejdsprocesser frem for i enkelte funktioner isoleret. Casene viser dermed, hvordan usability feedback kan fortolkes som relationel og strukturel viden gennem objektorienterede, relationelle og ontologiske perspektiver. Analysen understøtter dermed specialets argument om, at usability feedback kræver analytisk strukturering for at kunne indgå som grundlag for kvalificerede systembeslutninger i systemudvikling.

5.6 Konsekvenser for design og analyse af usability testing

Dette afsnit diskuterer, hvilke konsekvenser specialets analytiske perspektiv har for design og anvendelse af usability testing i systemudvikling. Formålet er ikke at foreslå en ny testmetode, men at diskutere hvordan usability testing kan tilrettelægges og analyseres, så usability feedback i højere grad kan indgå i analyse- og designarbejde. Med udgangspunkt i specialets relationelle forståelse af usability feedback bliver det centralt, at usability testing ikke alene fokuserer på enkeltstående problemer, men også på sammenhænge mellem bruger, system og kontekst.

5.6.1 Skift i testformål

Traditionel usability testing er ofte designet med fokus på at identificere konkrete problemer i brugerens interaktion med systemet, eksempelvis relateret til effektivitet, forståelighed og brugergrænsefladens udformning. Analysen i dette speciale peger imidlertid på, at usability feedback også kan anvendes som grundlag for indsigt i systemets struktur og sammenhænge. Et ændret testformål indebærer derfor, at usability testing ikke alene anvendes til at identificere, hvad der er problematisk for brugeren, men også til at undersøge hvorfor problemer opstår i systemets organisering og opbygning. Dette påvirker både valg af testopgaver, observationsfokus og analysearbejde. Usability testing fungerer dermed ikke kun som et evaluerende værktøj, men også som et analytisk redskab til at undersøge relationer mellem brugerhandlinger, systemfunktioner og kontekstuelle forhold.

5.6.2 Konsekvenser for testopgaver

Hvis usability testing skal bidrage til indsigt i systemets struktur, må testopgaverne designes, så de synliggør samspillet mellem systemets forskellige elementer. I stedet for isolerede opgaver, der tester enkelte funktioner, bør opgaverne tilrettelægges som sammenhængende arbejdsforløb på tværs af flere funktioner og systemområder. Denne type opgaver gør det muligt at analysere, hvordan navigation, informationsflow og systemrespons påvirker brugerens orientering og opgaveløsning. Fokus flyttes dermed fra spørgsmålet om, hvorvidt brugeren kan udføre en specifik handling, til hvordan brugeren forstår sammenhængen mellem systemets dele gennem opgaveløsningen.

Testopgaverne fungerer dermed ikke alene som et middel til at fremkalde usability-problemer, men også som analytiske greb til at identificere mønstre og strukturelle udfordringer i systemets organisering.

5.6.3 Konsekvenser for observation og analyse

Et strukturelt fokus i usability testing medfører også ændringer i, hvad der observeres, og hvordan observationerne analyseres. Observationen flytter sig fra primært at registrere individuelle fejl og misforståelser til at identificere mønstre i brugerens interaktion med systemet. Analysen retter sig dermed ikke alene mod enkelte usability-fejl, men mod hvordan systemets organisering og relationerne mellem funktioner, information og brugerhandlinger bidrager til de observerede problemer. Objektorienterede begreber som objekter, ansvar og relationer kan i denne sammenhæng anvendes som analytiske redskaber til at strukturere denne forståelse.

I et ontologisk perspektiv kan usability feedback organiseres som forbindelser mellem eksempelvis bruger, handling, funktion, problem og kontekst. Knowledge graphs kan i denne sammenhæng forstås som relationelle repræsentationer af disse forbindelser med henblik på at synliggøre mønstre på tværs af testsituationer. Formålet er ikke teknisk implementering, men at understøtte en mere systematisk og struktureret analyse af usability feedback. Denne tilgang øger samtidig kompleksiteten i analysearbejdet, fordi usability feedback ikke længere behandles som enkeltstående problemer, men som sammenhængende mønstre, der kræver både analytisk og domænemæssig forståelse.

5.6.4 Usability testing som strukturerende redskab

Denne tilgang ændrer ikke usability testings grundlæggende karakter, men forskyder dens rolle i systemudviklingsprocessen. I stedet for primært at blive anvendt sent i

udviklingsforløbet som evaluering af brugergrænseflader kan usability testing anvendes tidligere som grundlag for analyse og problemforståelse. Målet er ikke at omdanne usability testing til software testing eller underlægge den en verificerende logik. Det afgørende er derimod, at usability feedback fortolkes og struktureres, så den kan indgå mere direkte i systemudviklingens analyse- og beslutningsprocesser.

Objektorienteret analyse, ontologisk strukturering og relationelle repræsentationer fungerer her som analytiske mellemled, der understøtter arbejdet med at forstå systemets problemområder og sammenhænge.

6. Diskussion

Dette kapitel diskuterer specialets analytiske resultater med henblik på at vurdere deres teoretiske og praktiske implikationer. Med udgangspunkt i analysens centrale fund undersøges det, hvordan usability feedback kan forstås som en strukturel og relationel vidensform i systemudvikling. Kapitlet behandler først specialets analytiske og teoretiske bidrag, hvorefter forholdet mellem usability feedback, ontologisk strukturering og objektorienteret analyse diskuteres. Herefter behandles styrker, begrænsninger og analytiske spændinger ved den foreslåede tilgang samt implikationer for praksis og perspektiver for fremtidigt arbejde. Kapitlet afsluttes med en overgang til specialets konklusion.

6.1 Diskussion af specialets analytiske bidrag

Dette speciale har undersøgt, hvordan usability feedback fra brugertests kan analyseres og struktureres, så brugerindsigter i højere grad kan indgå i systemudviklingens analyse- og designarbejde. Hvor usability testing traditionelt anvendes som et diagnostisk værktøj til identifikation af problemer i brugerens interaktion med et system, har specialet argumenteret for, at usability feedback også kan bruges som en strukturel og relationel vidensform.

Specialets centrale analytiske bidrag består i introduktionen af objektorienteret analyse som et analytisk mellemled mellem usability testing og systemudvikling. Ved at anvende begreber som objekter, ansvar og relationer bliver det muligt at fortolke usability-problemer i relation til systemets opbygning frem for udelukkende som oplevelsesbaserede problemer på brugergrænsefladeniveau. Objektorienteret analyse anvendes dermed ikke som programmeringsmetode, men som en form for strukturel tænkning, der understøtter oversættelsen fra brugeroplevelser til strukturelle problemstillinger.

Specialet bidrager samtidig med et ontologisk og relationelt perspektiv på usability feedback. Gennem ontologisk strukturering og knowledge graph-inspirerede relationelle modeller forstås usability feedback ikke som isolerede observationer, men som relationer mellem bruger, handling, systemfunktion, problem og kontekst. Usability feedback fremstår dermed ikke som neutrale beskrivelser af systemet, men som analytiske fortolkninger af relationer mellem bruger, system og kontekst. Denne forståelse forskyder fokus fra enkelte usability-problemer til sammenhænge mellem systemets elementer.

Casene i analysen understøtter denne pointe. I guideline-casen opstod usability-problemer eksempelvis i relationerne mellem navigation, informationsstruktur og brugerens orientering

på tværs af flere informationskilder. I EHR-casen blev usability-problemer tæt knyttet til organisatoriske workflows, informationsflow og koordinering mellem systemets funktioner. Casene peger dermed på, at usability-problemer i komplekse systemer ofte ikke kan reduceres til enkelte funktioner, men opstår gennem relationer mellem flere systemelementer.

6.2 Usability feedback som strukturel og relationel viden

Et centralt argument i specialet er, at usability feedback ikke alene bruges som beskrivelser af individuelle brugeroplevelser, men som relationel viden om systemets struktur og anvendelseskontekst. Denne forståelse adskiller sig fra en mere traditionel usability-forståelse, hvor problemer primært identificeres som fejl eller mangler i brugergrænsefladen. Den ontologiske tilgang bidrager her med et begrebsligt perspektiv på usability feedback. Ved at identificere centrale begreber og relationer bliver det muligt at strukturere usability-viden mere systematisk. Ontologisk strukturering reducerer dermed ikke behovet for fortolkning, men gør fortolkningsprocessen mere eksplicit og systematisk. Problemer forstås ikke som objektive egenskaber ved systemet, men som relationelle fænomener, der opstår mellem bruger, system og kontekst. Dette stemmer overens med ontologiske perspektiver, hvor viden forstås gennem relationer mellem begreber og fortolkende strukturer frem for som isolerede fakta (Jakobsen & Graf, 2022).

Denne tilgang indebærer samtidig en vigtig analytisk forskydning. Hvor usability feedback traditionelt kan fremstå fragmenteret og situationsspecifik, muliggør ontologiske og relationelle modeller en analyse af mønstre og sammenhænge på tværs af observationer. Knowledge graph-perspektivet understøtter eksempelvis en forståelse af usability-problemer som relationelle forbindelser mellem systemets objekter, handlinger og informationsstrukturer.

Samtidig rejser denne tilgang også epistemologiske spørgsmål. Strukturelle modeller og ontologiske kategorier udgør ikke objektive repræsentationer af virkeligheden, men analytiske fortolkninger. Den relationelle strukturering af usability feedback eliminerer derfor ikke usikkerhed eller fortolkningsafhængighed, men flytter fortolkningsarbejdet fra observation til modellering. Forskellige analytikere kan dermed nå frem til forskellige strukturelle fortolkninger af den samme usability feedback. Dette peger på en grundlæggende analytisk spænding i specialets tilgang. På den ene side muliggør struktureringen en mere systematisk og sammenhængende analyse af usability feedback. På den anden side indebærer enhver strukturering en risiko for reduktion af kompleksitet, hvor nuancer i brugeroplevelser og kontekstuelle forhold kan gå tabt.

6.3 Objektorienteret analyse som analytisk mellemled

En central styrke ved objektorienteret analyse som analytisk mellemled er dens evne til at forbinde usability feedback med systemets strukturelle opbygning. Hvor usability testing producerer kontekstafhængig og fortolkningskrævende viden, og software testing bygger på en verificerende og kravbaseret logik, tilbyder objektorienteret analyse et strukturelt sprog til at arbejde mellem disse vidensformer. Ved at relatere usability-problemer til objekter, ansvar og relationer bliver det muligt at placere problemer i systemets struktur frem for alene i brugerens oplevelse. Dette kan understøtte mere kvalificerede beslutninger i analyse- og designarbejdet, fordi usability-problemer fortolkes i relation til systemets organisering frem for som isolerede interfaceproblemer.

Casene illustrerer denne pointe tydeligt. I guideline-casen kunne problemer med navigation og scrolling ikke forstås isoleret fra relationerne mellem forskellige informationskilder og navigationsstrukturer. I EHR-casen opstod usability-problemer i relationerne mellem workflows, informationsflow og koordinering af kliniske aktiviteter. I begge cases blev usability feedback dermed fortolket som strukturelle og relationelle problemstillinger frem for som enkeltstående fejl.

Objektorienteret analyse understøtter samtidig en mere helhedsorienteret forståelse af systemet. Relationer mellem objekter synliggør, hvordan ændringer i ét område kan påvirke andre dele af systemet. Dette kan potentielt styrke samarbejdet mellem UX-designere, systemudviklere og andre aktører ved at skabe et fælles analytisk sprog for forståelsen af usability-relaterede problemstillinger på tværs af udviklingsprocessen.

Samtidig er det vigtigt at understrege, at objektorienteret analyse ikke løser usability-problemer i sig selv. Tilgangen tilbyder en analytisk struktur til fortolkning og organisering af usability feedback, men afgør ikke, hvilke løsninger der er mest hensigtsmæssige. Objektorienteret analyse fungerer dermed som et refleksivt redskab snarere end en metode til automatisk problemløsning.

Samtidig adskiller specialets tilgang sig fra mere traditionelle usability-perspektiver, hvor usability feedback ofte omsættes til konkrete designændringer eller prioriteringer af funktionalitet. Formålet i dette speciale er ikke at erstatte sådanne tilgange, men at undersøge hvordan usability feedback kan analyseres som strukturel og relationel viden før den omsættes til konkrete løsninger eller krav. Objektorienteret analyse anvendes dermed ikke som en alternativ udviklingsmetode, men som et analytisk perspektiv til at understøtte fortolkningen af usability feedback i relation til systemets organisering og problemområder.

6.4 Begrænsninger og analytiske spændinger

Selvom den foreslåede tilgang tilbyder en strukturel måde at arbejde med usability feedback på, er der også væsentlige begrænsninger og analytiske spændinger forbundet med den. For det første forudsætter tilgangen analytiske kompetencer hos de aktører, der arbejder med usability feedback. Objektorienteret analyse kræver evnen til at abstrahere fra konkrete observationer og formulere strukturelle problemstillinger, hvilket ikke nødvendigvis er en del af alle UX-praktikeres eller systemudvikleres kompetenceprofil. Tilgangen stiller derfor krav om analytisk refleksion og fælles forståelse i udviklingsteamet.

Derudover indebærer tilgangen en risiko for overstrukturering. Hvis usability feedback for tidligt oversættes til faste strukturelle modeller, kan væsentlige nuancer i brugernes oplevelser og kontekstuelle forhold gå tabt. Strukturelle modeller kan skabe en oplevelse af orden og klarhed, selvom de underliggende problemer fortsat er komplekse, situerede og flertydige. Analysen peger desuden særligt på værdi i komplekse organisatoriske eller informationstunge systemer, hvor usability-problemer opstår på tværs af flere funktioner, workflows og informationsstrukturer. I mindre eller mere simple systemer kan den analytiske strukturering potentielt blive mere omfattende end den værdi, den tilfører.

Dette gælder også relationelle modeller såsom knowledge graphs. Selvom relationelle strukturer kan understøtte organiseringen af usability feedback og synliggøre sammenhænge mellem systemets elementer, indebærer de samtidig en risiko for overstrukturering og reduktion af kontekstuelle nuancer. Modelleringen af relationer beror fortsat på analytiske fortolkninger, og komplekse eller dynamiske brugssituationer kan være vanskelige at repræsentere fuldstændigt gennem faste relationelle strukturer.

Der eksisterer samtidig en spænding mellem strukturering og fleksibilitet. Jo mere usability feedback formaliseres gennem ontologiske kategorier og relationelle modeller, desto større er risikoen for, at brugeroplevelsens situerede karakter reduceres. Dette er særligt relevant i usability testing, hvor variation, fortolkning og kontekst netop udgør centrale datakilder. Endelig er objektorienteret analyse fortsat afhængig af fortolkning. Oversættelsen fra brugeroplevelse til objekter, ansvar og relationer er ikke neutral, men beror på analytiske valg. Tilgangen reducerer dermed ikke usikkerheden i usability feedback, men tilbyder et analytisk sprog og en struktur til at arbejde med denne usikkerhed.

6.5 Implikationer for praksis

Specialets resultater peger på flere implikationer for arbejdet med usability testing og systemudvikling i praksis. For systemudviklere indebærer perspektivet, at usability feedback kan anvendes mere direkte i analyse- og designarbejdet, hvis feedbacken struktureres analytisk. I stedet for primært at fungere som et sent kvalitetstjek kan usability testing anvendes tidligere i udviklingsprocessen som input til forståelsen af systemets struktur og problemområder.

For UX-designere peger perspektivet på en udvidet rolle, hvor arbejdet ikke alene handler om brugergrænseflader og interaktionsdesign, men også om at bidrage til forståelsen af systemets organisatoriske og strukturelle sammenhænge. Objektorienterede begreber kan i denne sammenhæng fungere som et fælles analytisk sprog mellem UX-designere og systemudviklere. Perspektivet kan samtidig have implikationer for samarbejde og beslutningstagning i udviklingsteams. Når usability feedback struktureres relationelt, bliver det lettere at diskutere problemer på tværs af fagligheder og relatere brugeroplevelser til systemets samlede arkitektur frem for til individuelle designønsker.

Samtidig indebærer tilgangen et øget behov for analytisk arbejde. Strukturering af usability feedback gennem ontologiske og objektorienterede perspektiver kræver tid, refleksion og fælles begrebsforståelse. I praksis må der derfor foretages en afvejning mellem den ekstra kompleksitet, som tilgangen introducerer, og den potentielle værdi, den kan skabe.

6.6 Perspektiver for fremtidigt arbejde

Dette speciale har haft et analytisk og teoretisk fokus på usability feedback, ontologisk strukturering og objektorienteret analyse som mellemled mellem usability testing og systemudvikling. Resultaterne peger på flere muligheder for videre arbejde. Et oplagt næste skridt er empirisk at undersøge, hvordan den foreslåede tilgang fungerer i konkrete udviklingsprojekter. Dette kunne eksempelvis ske gennem case-studier eller interventionsstudier, hvor usability testing kombineres med objektorienteret analyse i praksis.

Derudover kan videre forskning undersøge, hvordan konkrete analytiske artefakter såsom modeller, workshops eller værktøjer kan understøtte struktureringen af usability feedback i udviklingsteams. Specialet har primært arbejdet på et analytisk og begrebsligt niveau, mens operationalisering i praksis fortsat udgør et åbent spørgsmål.

Det vil desuden være relevant at undersøge, hvordan tilgangen kan kombineres med eksisterende udviklingsmetoder såsom agile processer, UX-workflows og requirements engineering. Her kunne fremtidig forskning bidrage med indsigt i, hvordan strukturel

usability-analyse kan integreres i etablerede udviklingspraksisser. Endelig kunne fremtidig forskning undersøge, hvordan knowledge graphs og ontologiske modeller mere konkret kan operationaliseres som digitale analyseværktøjer til organisering og visualisering af usability feedback.

7. Konklusion

Dette speciale har undersøgt, hvordan usability feedback fra brugertests kan analyseres og struktureres, så usability-viden kan indgå mere direkte i systemudviklingens analyse- og designarbejde, samt hvilken rolle objektorienteret analyse og design kan spille som bindeled mellem usability testing og software testing. Analysen har vist, at usability testing og software testing bygger på forskellige testlogikker og kvalitetsforståelser. Hvor software testing er verificerende og kravbaseret, producerer usability testing kontekstafhængig og kvalitativ viden om brugernes interaktion med systemet. Dette skaber et spændingsfelt i systemudvikling, hvor usability feedback kan være vanskelig at integrere direkte i analyse- og designprocesser.

Specialet har argumenteret for, at usability feedback kan bruges som relationel viden om sammenhænge mellem bruger, handling, systemfunktion og kontekst frem for som isolerede interfaceproblemer. Gennem ontologiske og relationelle perspektiver bliver det muligt at analysere usability-problemer som strukturelle sammenhænge i systemets organisering. Specialets centrale bidrag er introduktionen af objektorienteret analyse som et analytisk mellemled mellem usability testing og systemudvikling. Ved at anvende begreber som objekter, ansvar og relationer kan usability feedback fortolkes i relation til systemets strukturelle opbygning frem for alene som individuelle brugeroplevelser.

Casene i analysen understøtter denne pointe ved at vise, hvordan usability-problemer i komplekse systemer ofte opstår i samspillet mellem informationsstrukturer, workflows, navigation og organisatoriske processer frem for i enkelte funktioner isoleret. På denne baggrund konkluderer specialet, at usability feedback kan analyseres og struktureres gennem ontologiske, relationelle og objektorienterede perspektiver, som understøtter en mere systematisk forståelse af systemets problemområder. Objektorienteret analyse og design kan dermed fungere som et analytisk bindeled mellem usability testing og software testing ved at forbinde usability-indsigter med mere strukturerede analyse- og udviklingsprocesser.

Specialet har haft et teoretisk og analytisk fokus, og der er derfor behov for fremtidigt empirisk arbejde, der undersøger, hvordan den foreslåede tilgang kan anvendes i konkrete udviklingsprojekter og organisatoriske praksisser. Specialet peger dermed på, at usability feedback ikke blot bør forstås som evaluering af brugergrænseflader, men som en potentiel kilde til strukturel indsigt i systemers organisering og anvendelseskontekst.

Litteraturliste

Edwards, P. J., Moloney, K. P., Jacko, J. A., & Sainfort, F. (2008). Evaluating usability of a commercial electronic health record: A case study. *International Journal of Human-Computer Studies, 66*(10), 718-728. https://doi.org/10.1016/j.ijhcs.2008.06.002

Guarino, N., Oberle, D., & Staab, S. (2009). What is an ontology? In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (pp. 1-17). Springer. https://doi.org/10.1007/978-3-540-92673-3_0

Hertzum, M. (2016). Usability testing: Too early? Too much talking? Too many problems? *Journal of Usability Studies, 11*(3), 83-88.

Hertzum, M., & Clemmensen, T. (2012). How do usability professionals construe usability? *International Journal of Human-Computer Studies, 70*(1), 26-42. https://doi.org/10.1016/j.ijhcs.2011.08.001

Hertzum, M., Hansen, K. D., & Andersen, H. H. K. (2009). Scrutinising usability evaluation: Does thinking aloud affect behaviour and mental workload? *Behaviour & Information Technology, 28*(2), 165-181. https://doi.org/10.1080/01449290701773842

Hogan, A., Blomqvist, E., Cochez, M., D'Amato, C., de Melo, G., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. *ACM Computing Surveys, 54*(4), 1-37. https://doi.org/10.1145/3447772

Jakobsen, D., & Graf, S. (2022). Formal, measurable ontologies for worldviews. In R. Polovina, S. Polovina, & N. Kemp (Eds.), *Measuring ontologies for value enhancement: Aligning computing productivity with human creativity for societal adaptation* (pp. 86-97). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-22228-3_5

Khodambashi, S., & Nytrø, O. (2016). Usability evaluation of published clinical guidelines on the web: A case study. *Proceedings of the IEEE International Symposium on Computer-Based Medical Systems*, 140-145. https://doi.org/10.1109/CBMS.2016.11

Li, H., Appleby, G., Brumar, C. D., Chang, R., & Suh, A. (2024). Knowledge graphs in practice: Characterizing their users, challenges, and visualization opportunities. *IEEE Transactions on Visualization and Computer Graphics, 30*(1), 1-11. https://doi.org/10.1109/TVCG.2023.3326904

Mathiassen, L., Munk-Madsen, A., Nielsen, P. A., & Stage, J. (2018). *Object-Oriented Analysis & Design* (2nd ed.). Metodica ApS.

Stapp, L., Roman, A., & Pilaeten, M. (2024). *ISTQB® certified tester foundation level: A self-study guide syllabus v4.0*. Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42767-1