
Interaksjonsdesign som prosess 2

TDAT2003 Systemutvikling 2 med web-applikasjoner
Institutt for informatikk og e-læring
Anette Wrålsen september 2016



Innholdsfortegnelse

Evaluering av brukergrensesnitt	1
Hvorfor evaluere?	2
Evaluering vs. testing	3
Evaluering som involverer brukere	3
Klassiske brukertester	3
<i>Planlegging</i>	3
<i>Testplanen</i>	4
<i>Samtykke- og taushetserklæring</i>	5
<i>Oppgavene</i>	5
<i>Gjennomføring</i>	6
<i>Vær klar over brukertestingens begrensninger</i>	6
A/B-tester (sammenligningstester)	7
Feltstudier i naturlige omgivelser	8
Opportunistiske evalueringer	8
Evaluering som ikke involverer brukere	8
Heuristiske gjennomganger	8
Pluralistiske gjennomganger	9
Prediktive modeller	9
Noen evalueringsmetoder som er spesielt relevante for nettsteder	9
Evaluering i praksis	11
Oppsummering	12
Referanser	12

Evaluering av brukergrensesnitt

Denne leksjonen handler om evaluering. På samme måte som vi bør designe ethvert aspekt av et brukergrensesnitt nøye, kan ethvert aspekt evalueres. Du kan teste fargevalget, du kan teste

enkelte skjermbilder, du kan teste hvor lang tid det tar å utføre viktige oppgaver, og så videre. Det å utføre tester er ofte dyrt og tidkrevende, og det betyr at du må prioritere, og dermed velge ut hva som er de viktigste aspektene ved systemet. Men du må ikke hoppe over denne delen, uten noen form for evaluering har du egentlig ingen anelse om kvaliteten på produktet du utvikler. En ting er den konseptuelle modellen du lager, men hvis du ikke er sånn noenlunde sikker på at de fleste brukere vil ende opp med lignende konseptuelle modeller når de bruker produktet, er hele designfasen ganske meningsløs.

Evaluering står som siste trinn i IxD-prosessen, men det er ingen grunn til ikke å starte evaluering tidligere. Man kan kjøre enkle tester allerede på den konseptuelle modellen man lager (F.eks.: Klarer personer som ikke er involvert i utviklingen forstå hvordan ting henger sammen hvis du bare presenterer dem for konseptene og metaforene og ikke forklarer detaljer i hvordan ting henger sammen?) og på prototypene. Dette kalles ofte for *formativ evaluering*¹³, dvs. at man gjør en evaluering for å påvirke utviklingen og sikre at designene man jobber med oppfyller kravene. I tillegg har vi såkalt *summativ evaluering*¹⁴, som er en evaluering av sluttproduktet (når vi ikke lenger kan gjøre endringer).

Så, for å understreke hvor viktig evaluering egentlig er, dedikerer vi en hel leksjon til dette temaet. De fleste metodene vi skal innom kan brukes for de fleste systemer, mens noen vil være spesielt rettet mot nettsteder.

Hvorfor evaluere?

Før vi går videre oppsummerer vi noen av grunnene til at evaluering er viktig:

- ✓ **Evaluering avslører feil.** Det er nesten umulig å ikke gjøre noen feil, og tidlig testing er rett og slett en enkel og ofte billig måte å oppdage disse på.
- ✓ **Evaluering hjelper deg med å oppdage ting du ikke har tenkt på.** Designere er ikke brukere, og uansett hvor mye arbeid du legger ned i designfasen er det alltid noen som vil bruke produktet ditt på en helt annen måte enn du har tenkt. Dette er det viktigste poenget med brukertesting – å oppdage sider ved designet ditt du ikke har tenkt på selv, men som viser seg å bli problematisk for brukerne.
- ✓ **Evaluering viser hva brukerne faktisk gjør, og ikke hva du eller de tror de kommer til å gjøre.** Du kan la brukere synse så mye de vil om hva de ønsker ut av et system eller hva de vil gjøre i en gitt situasjon, men praksis er noe helt annet. Vi ofte alt annet enn rasjonelle valg når vi skal løse problemer. Vi prøver å være raske og leser ikke all informasjonen som er tilgjengelig i et grensesnitt, vi misforstår metaforer og vi gjør ting slik vi er vant til det i stedet for slik det nye systemet legger opp til det. Med andre ord, vi må undersøke hva som skjer i praksis når brukere anvender systemet vårt, det holder ikke med teorien. Evaluering gir oss mange verktøy for å undersøke dette.
- ✓ **Evaluering hjelper deg å holde fokus på brukerne.** Det er lett å miste brukerne av syne når man er opptatt av sitt eget design. Mange, kanskje spesielt oppdragsgivere, opplever brukertesting som dyrt og mener at designerne bør klare å finne svakhetene med designet sitt selv. Men det viser seg stadig at det lønner seg å satse på god brukskvalitet, og da er

I Jakob Nielsens [Alertbox fra juli 2012](#), argumenterer han for at hvis du bare har råd til *en* aktivitet for å bedre brukskvaliteten i et prosjekt, bør dette være nettopp brukertester.

¹³ Dette svarer til *underveisvurdering* i læringssammenheng, der man får tilbakemeldinger underveis for å kunne forbedre læringsprosessen.

¹⁴ Dette svarer til *sluttvurderingen* i læringssammenheng – for eksempel eksamenskarakteren du sitter igjen med.

brukertesting et av de viktigste virkemidlene (i tillegg til å følge etablerte retninglinjer). Jo større og mer mangfoldig brukergruppe du har, jo viktigere er dette.

Evaluering vs. testing

Til slutt en liten kommentar om språkbruk: Du har kanskje lagt merke til at jeg bruker begrepene *evaluering* og *testing* tilsynelatende om hverandre. I utgangspunktet tenker jeg her på evaluering som et bredt begrep, som inkluderer det vi typisk kaller tester, men også andre ting, som å bruke formelle modeller, ekspertgjennomganger eller å gå gjennom sjekklister over typiske feil (ting som de fleste ikke vil kalle tester i ordets reneste forstand). Dette er altså en leksjon om *evaluering*, der vi blant annet skal innom mange typer *tester*.

Evaluering som involverer brukere

I denne delen skal vi se på konkrete måter å evaluere på. Vi kommer til å følge inndelingen fra over, men merk at noen metoder kan passe inn i flere av kategoriene alt ettersom hvordan vi gjennomfører evalueringen.

Klassiske brukertester

Dette er tester der vi typisk inviterer brukere til å teste ut en prototype av det nye systemet (eller det eksisterende systemet om vi ønsker å finne ut hva som er mest problematisk med dagens design), og samler inn data om hvordan det går som vi så bruker til å finne designfeil, ta avgjørelser og forbedre designet vårt. Det er også vanlig å kjøre samme test flere ganger, først for å finne problemer, og senere for å bekrefte at det nye designet har færre problemer (og helst ikke har introdusert nye feil).

Den vanligste måten å gjøre denne typen tester for nettstedet på er rett og slett å få tak i en håndfull personer i målgruppen og be dem løse en oppgave i systemet som skal testes mens de tenker høyt. Dette er en i utgangspunktet enkel metode som kan gjøres systematisk og metodisk strengt, eller relativt uformelt. Uansett får vi informasjon fra det (selv om kvaliteten på informasjonen henger sammen med kvaliteten på testen). I følge NetLife Research¹⁵ gir brukertesting av nettstedet deg både konkret kunnskap om hvordan nettstedet fungerer, du får godt grunnlag for diskusjon og avgjørelser om nettstedet, og du finner ut om funksjonaliteten du tilbyr faktisk er den brukerne ønsker seg. Dette gjelder også for datasystemer generelt.

Vi skal nå se på hva vi må tenke på og hvordan vi bør gå fram for å gjennomføre en brukertest.

Planlegging

Det er flere ting du bør tenke gjennom før du skal gjennomføre en brukertest:

- ✓ **Hva er det du skal teste, og hvordan skal du gjøre det?** Du bør være konkret på hva du ønsker å finne ut av. Skal du teste en bestemt funksjonalitet? Fargevalget i brukergrensesnittet? Hvor lang tid som går med til å gjøre en oppgave? Layout? Hvor gode menyene er? Testen må så legges opp etter dette. En test som bare stiller tilfeldige spørsmål gir deg bare tilfeldige svar. Du bør også selvsagt velge ut hva som skal testes og hvordan du går fram ut fra *hva som er de viktigste egenskapene mhp. brukskvalitet og brukeropplevelsen* med akkurat dette systemet.

¹⁵ <http://www.iallenkelhet.no/gjør-det-selv-5-trinn-til-brukertestet-nettsted>

-
- ✓ **Hvem skal være testbrukerne, hvor mange brukere skal du ha og hvordan skal du rekruttere dem?** Du må finne en måte å få tak i brukere i målgruppene på, og du må avgjøre om du skal belønne dem for deltakelse eller ikke.
 - ✓ **Hva skal du måle, hvordan skal du samle inn informasjon under testen, og hvordan skal du analysere den etterpå?** Her er det mange muligheter. Du kan gjøre notater, du kan ta opp video eller lyd, du kan ta opp skjermen til brukeren mens du gjør notater, du kan ta vare på logger, du kan intervju brukerne etter på, du kan lage spørreskjemaer som skal fylles ut før og/eller etter testen.
 - ✓ **Når og hvor skal testen være, og hvem skal gjennomføre den?** Slike praktiske ting er en viktig del av planleggingen. Har du egnede lokaler? Skal dere leie hos noen andre, og kanskje til og med leie eksperttestere hos et firma som spesialiserer seg på dette?

Akkurat hvor mange testbrukere du bør la gjennomføre testen er det ikke full enighet om, men de fleste er enig i at du kan få veldig mye god informasjon fra bare 5-10 personer. Grunnen til at du ikke trenger så mange er at brukere generelt gjør ting relativt likt, og selv om to brukere ikke gjør ting helt likt så er mengden *ny* lærdom du kan få fra en testbruker ekstra begrenset når du allerede har testet ti personer. Selvsagt er det mulig at du har gjort et veldig lite representativt utvalg, men velger du testbrukere med litt omhu burde du få mye nyttig informasjon. Merk også at om du bare får tak i ti personer totalt er det kanskje bedre å kjøre to forskjellige tester med fem personer i hver, enn å teste det samme på alle ti!

For å få rekruttert brukere kan du gå fram på mange måter, og det er relativt vanlig å betale brukere for å delta. Er det veldig vanskelig å få tak i folk, så prøv å få gjort tester allikevel, om ikke annet så på slekt og venner. Du kan fortsatt få nyttige resultater ut av testen. Husk også at det er lurt å velge testbrukere som ikke er eksperter, dvs. som ikke har altfor mye kjennskap til data og/eller lignende systemer fra før, det er mye større sannsynlighet for at disse brukerne finner faktiske *brukskvalitetsfeil* med systemet og ikke bare peker på ting de ikke *liker* (ut fra erfaring med andre systemer).

Testplanen

Når du planlegger en brukertest må du i følge *Praktisk Brukertest* av Eli Toftøy-Andersen og Jon Gunnar Wold (en bok som for øvrig anbefales varmt – den gir deg en lettlest, konkret og praktisk innføring i brukertest med mange eksempler) ta stilling til følgende spørsmål:

1. **Hva er formålet med brukertesten?** Et klart formål gjør det lettere å designe testen og lettere å tolke resultatene etterpå.
2. **Hvilken funksjonalitet skal testes?** Dette bestemmer du med utgangspunkt i svare på spørsmålet over. Test gjerne både om brukerne finner fram til funksjonaliteten og om de finner ut av den (gjørne med forskjellige problemer som skal løses med samme funksjoner).
3. **Hva slags system skal du teste?** Det kan virke åpenbart hva som er svaret her, nemlig systemet du lager. Men underveis i utviklingen forekommer jo dette i mange versjoner. Skal du teste en prototyp? I så fall hvilken? Eller skal du teste en tidlig iterasjon av det endelige systemet?
4. **Hva slags personer skal være testbrukere?**
5. **Hvor skal testen foregå?** Her er det to hovedtyper svar: I deres lokaler, tilrettelagt for testen, eller i brukeres naturlige miljøer (for eksempel på kontoret der systemet skal brukes, ute på sightseeing hvis du lager en mobilapp for turister osv.).
6. **Hva slags testutstyr skal brukerne benytte?** Du bør ha ekstra utstyr tilgjengelig, både fordi utstyr kan gå i stykker og fordi noen brukere insisterer på å bruke bestemte former for datautstyr (“-Jeg kan ikke bruke touchpad, må ha mus, jeg skjønner du, ellers får jeg så vondt i skulderen.”). Utstyr inkluderer her programvare. Med mindre det er et poeng at

brukere bruker en bestemt nettleser, vurder å ha flere nettlekere tilgjengelig til test av et nettsted, og la brukerne velge den de helst vil ha.

7. **Hvilke oppgaver skal brukerne få?** 5-10 konkrete oppgaver, valgt ut fra hva som skal testes, er ofte passelig. Ha klare kriterier for når en oppgave regnes som løst (eller hva som skal måles, f.eks. tiden det tar å løse oppgaven).
8. **Hvilken timeplan skal du følge?** Timeplan er viktig, både for deg og for brukerne som venter på tur. Beregn god tid pr. bruker, og litt ekstra tid i mellom til etterarbeid/forberedelser til neste mann og annet som dukker opp.
9. **Hvilke spørsmål skal du stille brukerne før og etter testen?** Du bør benytte deg av sjansen til å stille demografiske spørsmål før testen, og kanskje noen spørsmål om brukeropplevelsen etterpå. Skal brukeren svare på en spørreundersøkelse, bør denne legges ved dokumentet.
10. **Hvem er med i testteamet?** Det er ofte mye arbeid som ligger i en slik test, og den utvikles best i samarbeid med flere. Man må planlegge innhold, rekruttere brukere, lage oppgaver og intervju-spørsmål/undersøkelse, gjennomføre (det tar gjerne en arbeidsdag å gjennomføre fem brukertester), analysere informasjon og planlegge hvordan man skal bruke resultatet til å forbedre systemet (eller begrunne at det allerede er perfekt om du er veldig heldig).
11. **Hvordan skal vi formidle funnene i testen?**
12. **Når skal vi møtes for å definere videre aksjonspunkter på bakgrunn av resultater fra testen?**

Svarene på disse spørsmålene bør oppsummeres i et dokument, nemlig *testplanen*. God planlegging gir gode tester, som igjen gir nyttige resultater. Det er viktig å ha planer både for forberedelser, gjennomførelse og etterarbeid. Hva du gjør på de forskjellige trinnene påvirker resultatet, og uten en gjennomtenkt plan blir resultatet mer tilfeldig og mindre nyttig. Punkt 12 på lista over er veldig viktig – hvis vi ikke skal bruke resultatene til noe, hvorfor gjennomføre testen i det hele tatt?

Samtykke- og taushetserklæring

Husk også at du bør lage en samtykke- og taushetserklæring som brukerne undertegner når de er med i testen. Dette er viktig både for brukerne og utviklerne:

- ✓ Brukerne bør få vite hvilke data som samles inn – observasjoner, video, lyd, intervjudata osv.
- ✓ Brukerne bør forsikres om at resultatene fra testen behandles konfidensielt.
- ✓ Utviklerne/eier ønsker gjerne at brukerne signerer en taushetserklæring om systemet, siden det fortsatt er under utvikling.

Erklæringen bør også inneholde informasjon om hvilken kompensasjon brukeren eventuelt får for å delta.

Oppgavene

Å lage oppgavene er en veldig viktig del av testen. Oppgavene bør være basert i formålet og funksjonaliteten som skal testes. De kan gi brukeren varierende grad av frihet:

- ✓ **Helt konkrete oppgaver:** Brukeren skal løse et spesifikt problem, og vi har klare kriterier for når oppgaven regnes som løst. Gjør det enkelt å sammenligne hvordan forskjellige brukere gjør det. Ulempen er at du leder brukeren kanskje litt vel mye, slik at hun kan ende opp med å gjøre ting på en annen måte enn hun ville gjort ellers ("Bruk funksjon A til å løse problem B").
- ✓ **Delvis åpne oppgaver:** Brukeren får oppgaver, men ikke klare retningslinjer for hvordan de skal løses ("Løs problem B"). Dette gjør det vanskeligere å sammenligne brukere og stiller

høyere krav til den som observerer testen når det gjelder å få med seg det som skjer, men gjør at brukeren kanskje oppfører seg mer “naturlig”.

- ✓ **Helt åpne oppgaver:** Her gir du brukeren bare noen få, vage føringer, og ser hvor det hele ender. For eksempel: Du skal kjøpe ny bil og vil se om du finner noe interessant på internett, hvordan gjør du det? Den store åpenbare ulempen her er at du ikke kan styre hva som skjer i det hele tatt. Kanskje finner brukeren aldri fram til ditt nye nettsted bilersomgjøratdusmiler.no i det hele tatt?

Ellers kan vi merke oss at kategoriene over svarer godt til typene intervjuer vi så på tidligere (strukturerte, semi-strukturerte og ustrukturerte) og har lignende fordeler og ulemper!

Gjennomføring

Testen kan som sagt både foregå under strengt kontrollerte forhold hos dere, eller i mer naturlige omgivelser. Fordelen med førstnevnte er selvsagt at du har alle muligheter til å kontrollere omgivelsene – forhindre forstyrrelser, forsikre deg om at alt relevant utstyr er tilgjengelig og sikre at forskjellige brukere har likest mulig forutsetninger. Ulempen er at situasjonen kan føles unaturlig for brukeren, og øke sannsynligheten for at brukeren blir nervøs eller gjør ting på en måte han ikke ellers ville gjort. Prosedyren som følger tar utgangspunkt i at du skal ha test under kontrollerte forhold hos dere, ellers må den tilpasses stedet testen skal foregå.

- ✓ Gjør klar et stille rom med datamaskin (eller annet utstyr om det er relevant) klargjort, samt alt av dokumenter inkludert samtykke- og taushetserklæringer.
- ✓ Planlegg god tid, og bruk tid på å forklare brukeren hva som skal skje. Ha gjerne klart en tidsplan som brukeren kan få se.
- ✓ Gjennomfør selve testen.

Testen kan gjennomføres på flere måter. Det vanligste er å gi brukeren oppgavene, gjerne skriftlig, og be ham tenke høyt mens han løser dem og du noterer deg viktige ting som skjer (Hvilke menyer undersöker han? Prøver han flere måter før han finner den riktige? Hvor lang tid tar det?). Hvis du har avtalt det på forhånd, kan du gjerne ta opp video og/eller lyd som kan analyseres i ettertid. Noen brukere synes det er unaturlig å tenke høyt, da kan du velge å stille dem enkle spørsmål for å hjelpe dem med høyttenkingen, men uten å lede dem i en bestemt retning (Hva forsøker du å gjøre nå? Skjedde det du trodde skulle skje når du klikket på det menyvalget?)

Du bør ha bestemt på forhånd i hvilken grad du skal hjelpe brukeren underveis. Skal du la henne klare seg helt alene? Eller skal du hjelpe litt til hvis hun blir stående helt fast? I virkeligheten vil jo ikke brukeren ha en person der til å hjelpe seg når hun blir stående fast så dette er kanskje den mest realistiske situasjonen, men samtidig risikerer du å ikke få testet viktige deler av systemet hvis brukeren ikke finner fram til funksjonene du ønsker å undersøke.

Når du observerer og gjør notater, husk å gjøre nettopp det – observér og ta notater! Skriv objektivt ned hva brukeren gjør, ikke gjør vurderinger eller foreslå alternative løsninger. Vær også forberedt på å få knust noen illusjoner. Det vil helt sikkert være deler av designet som ikke fungerer slik du trodde og designvalg du var fornøyd med som brukere rett og slett ikke liker. Da gjelder det å huske at det var nettopp dette som var målet med testingen, og at du nå får muligheten til å lage et enda bedre design!

Vær klar over brukertestingens begrensninger

Brukertester som de over er i stor grad *kvalitative* vesener. Det betyr at de først og fremst lærer deg om hvordan brukere opplever og løser oppgaver i systemet ditt og hvor og hvorfor det stopper opp for dem, men i mindre grad kan svare på mer *kvantitative* spørsmål som “bør denne knappen være grå eller blå?”. For å få svar på det bør du bruke metoder av mer kvantitativ natur (som for eksempel A/B-tester, beskrevet under). Altså: brukertester avslører viktige brukskvalitetsproblemer

og feil og gir deg nye perspektiver på designet ditt, men de kan vanligvis ikke si med sikkerhet hvordan *alle* brukere eller *brukere flest* opplever systemet ditt.

- ✓ Mange flere eksempler: <http://www.abtests.com>
- ✓ Jakob Nielsen om A/B-testingens rolle: <http://www.useit.com/alertbox/20050815.html>
- ✓ Eksempler og innsikter fra det norske firmaet NetLife Research: <http://www.iallenkelhet.no/5-eksempler-pa-ab-testing-som-bor-gi-deg-lyst-til-a-prve-selv>
- ✓ Gjør din egen A/B-testing med Google Analytics Content Experiments: [Video](#) og [informasjon](#).

Til slutt noen lenker som du bør se på før du går videre:

- ✓ Lite videoeksempel på kvalitativ brukertest med papirprototype, mange gode tips: <http://www.youtube.com/watch?v=9wQkLthhHKA>
- ✓ Eksempel på en brukertest utført av det danske firmaet DialogDesign. Testingen ble gjort i 2001 for den amerikanske nettbutikken Tower Records (som selger musikk og video), og den komplette rapporten fra testingen er lagt ut her: http://www.dialogdesign.dk/Test_Reports.htm
Legg spesielt merke til appendiksene, som beskriver hvordan testingen ble gjennomført.

A/B-tester (sammenligningstester)

Dette er en kvantitativ testmetode der man lager to (eller flere) varianter av et design, presenterer brukere for en tilfeldig variant og rett og slett ser hvilken variant som har best resultat (ut fra et eller annet forhåndsbestemt mål). Metoden er blant annet mye brukt på nett, og er en konkret og grei metode å bruke for nettstedet som har en viss mengde trafikk der man har gode måter å måle suksessen til et design på (f.eks. i antall klikk, antall salg eller lignende). Legg merke til hvordan denne skiller seg fra klassiske brukertester, som vanligvis er *kvalitative* av natur. Vi spør ikke lenger hva brukeren synes og tenker, men observerer hvordan han eller hun handler.

Situasjoner som egner seg for A/B-testing er situasjoner der du lett kan vise fram alternative design for brukere, der man ønsker å gjøre konkrete og lett målbare designendringer og der det er lett å måle resultatet av endringene. Lett målbare designendringer kan være skriftstørrelse, farger, utforming av grafikk m.m. Dette er fordi mer komplekse endringer kan gi mer komplekse effekter, og det kan da bli vanskelig å si noe om den eksakte årsaken til resultatene av testen. Ønsker vi dermot bare å få bekreftet at et design fungerer bedre enn et annet, kan vi selvsagt ha mer komplekse designendringer. Da må vi være obs på at resultatene kanskje ikke er overførbare til andre situasjoner og senere designvalg (så det er begrenset hva vi kan lære av dem), men dette kan ofte være akseptabelt.

Merk at vi må ha en konkret måte å måle hvilket design som er best på, ellers er denne metoden nesten uten hensikt.

Dette er en relativt billig måte å teste på, gitt at det ikke er for krevende å lage de to designvariantene. I følge Jakob Nielsen er dette særlig en nyttig metode i situasjoner der kjente retningslinjer ikke kan si noe sikkert om hvilket av to design som er best.

For å få meningsfulle målinger bør begge designene være tilgjengelige over et visst tidsrom, helst minst en måned. Noen ganger tester man sekvensielt, dvs. at først ligger et design ute og senere et annet. Da bør man være obs på at det kan være andre ting enn bare designet som påvirker resultatet – for eksempel selger de fleste nettbutikker mer når det nærmer seg jul uansett design. Her er flere lenker om A/B-testing:

Merk at denne metoden kan brukes både i kontrollerte omgivelser og i naturlige omgivelser (typisk ved å servere forskjellige alternativer av nettsteder), og både tidlig og sent i utviklingsprosessen.

Feltstudier i naturlige omgivelser

Man kan også gjøre andre typer studier i mer naturlige omgivelser. Dette vil som regel innebære at utviklerne drar ut i felten, og lar brukerne anvende systemet eller en prototyp på systemet der. Man lar da typisk brukere få en utgave av systemet, la dem bruke det fritt et par uker og intervjuer dem etterpå samtidig som man prøver å observere uten å påvirke brukerne på noe som helst vis. Man kan også velge å bare logge trafikk og bruk uten å observere personlig.

Det å gi brukere en eksperimentell versjon av systemet og la dem teste den over tid kan være en god måte å gjøre *mulighetsstudier*, forstå kravene til systemet eller undersøke nye anvendelser av eksisterende teknologi på.

Generelt er denne typen brukerstudier vanskelige og dyre å gjennomføre, men det lar oss lære ting som klassiske brukertester ikke kan svare på. Hvis vi velger å ikke observere personlig, er vi også avhengige av hva som blir logget og/eller hva brukerne forteller i intervjuer for å lære.

Opportunistiske evalueringer

Dette er enkle tester som typisk gjøres tidlig i utviklingen, der man bare haker tak i noen få brukere og ber dem gi rask tilbakemelding på en konkret idé. Dette er ofte veldig nyttige og billige tester, som kan brukes til å avgjøre om et design bør tas videre til en prototype eller droppes.

Evaluering som ikke involverer brukere

Dette er evalueringsmetoder der eksperter og utviklere går gjennom designet på forskjellige måter. Dette er billig og enkelt, men resultatet avhenger veldig av kompetansen til de som går gjennom designet. Det er også farer for at ekspertene går i samme felle som ekspertbrukerne: De finner "feil" som egentlig bare er ting de selv ikke liker eller mener vil bli et problem, uten at det egentlig vil bli et problem i praksis. Dette problemet kan hjelpes ved å bruke flere eksperter, slik at man får flere perspektiver.

Siden dette faget fokuserer spesielt på webutvikling, ser vi også nærmere på tester som er spesielt relevante for nettsteder (inkludert enkelte typer tekniske tester).

Heuristiske gjennomganger

Dette handler om å gå gjennom velkjente tommelfingerregler og retningslinjer for brukskvalitet. Slike finnes det mange av (som f.eks. den i første IxD-øving). Man kan også ta utgangspunkt i kunnskap om hva brukere typisk finner problematisk med den aktuelle typen systemer. For nettsteder finner du mange retningslinjer i en av leksjonene i dette faget!

Som antydnet over bør man la flere eksperter se på systemet med utgangspunkt i de samme retningslinjene, for så å møtes og bli enige om en felles rapport. Denne typen gjennomganger er en fin måte å utfylle brukertester på, da de ofte finner andre problemer enn de som kommer fram på brukertestene.

Pluralistiske gjennomganger

Her møtes brukere, utviklere og brukskvalitetseksperter for å jobbe seg gjennom viktige scenarioer

- ✓ HTML-validator: <http://validator.w3.org/>
- ✓ CSS-validator: <http://jigsaw.w3.org/css-validator/>
- ✓ Feed-validator: <http://validator.w3.org/feed/>
- ✓ Lenke-validator: <http://validator.w3.org/checklink/>

og diskutere problemer som kan oppstå. Typisk vil en av deltakerne spille rollen som bruker i scenarioet. Dette er en god måte å involvere brukere på gruppenivå, samt at det gir et møtepunkt mellom utviklere og brukere (som vi tidligere har påpekt at ofte er et mål i seg selv).

Prediktive modeller

Dette er et tema vi ikke skal komme inn på her, men vi vil allikevel nevne at MMI-feltet byr på mange modeller som prøver å forutsi hvordan det vil være å bruke et gitt brukergrensesnitt. For eksempel har vi Fitts' lov¹⁶, som er en matematisk lov som forutsier hvor lang tid det vil ta å flytte en musepeker fra et sted til et annet sted på en skjerm.

Noen evalueringsmetoder som er spesielt relevante for nettsteder

Her er en oversikt over noen typer evalueringer som er viktige å gjøre når du utvikler et nettsted:

- ✓ **Kodevalidering.** Å *validere* kode er rett og slett å sjekke at koden er korrekt og følger anbefalte standarder. W3C tilbyr valideringsverktøy både for (X)HTML-kode, CSS-kode og feeds (RSS og Atom), og en tjeneste som sjekker at lenkene på en nettside faktisk fører noe sted (for å unngå såkalte *døde lenker*, lenker som peker mot nettsteder som ikke lenger eksisterer). Her er lenker til validatorene:

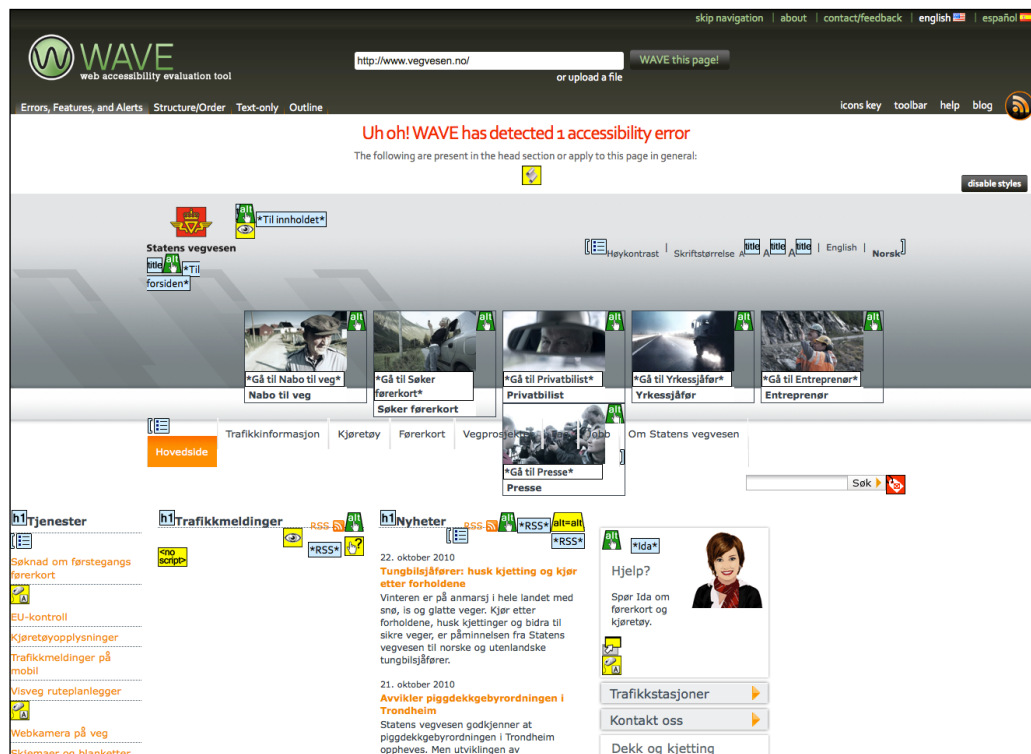
Her finner du mange typer sammenligninger av nettlesere:
http://en.wikipedia.org/wiki/Comparison_of_web_browsers

Du kan teste et hvilket som helst nettsted, prøv selv og se hva du får som resultat! Husk at det å følge standarder er viktig, både for å gjøre koden lettere å skrive og tolke for mennesker, fordi det sikrer at koden tolkes riktig i flest mulig nettlesere, og fordi det gir bedre universell utforming.

- ✓ **Testing i forskjellige nettlesere og på forskjellige plattformer.** Dessverre tolker ikke alle nettlesere kode likt, og selv om dette blir et stadig mindre problem da flere og flere nettlesere begynner å følge W3Cs anbefalinger, er det fortsatt nødvendig å teste nettstedet ditt i flere nettlesere. Det finnes flere tilbydere av denne typen tester på nett, i tillegg til spesialisert programvare.

¹⁶ [Fitts' lov](#)

- ✓ **Testing av fleksibiliteten til et design.** Fleksibilitet betyr her hvordan designet endrer seg med forskjellige oppløsninger, sidebredder og fontstørrelser. Hva skjer når brukeren halverer vindusbredden eller har skjerm med lav oppløsning? Eller når brukeren endrer fontstørrelsen? Uforutsette ting kan skje (alt etter hvor solid CSS-koden er), så dette bør vi teste regelmessig under utvikling.
- ✓ **Testing av grad av universell utforming og brukskvalitet.** Dette bør vi selvsagt alltid undersøke på mange forskjellige måter, både for nettstedet og andre typer systemer, men for nettsteder finnes det gratis verktøy som gjør dette for deg automatisk som f.eks. WAVE (<http://wave.webaim.org/>). Her er resultatet av å kjøre vegvesen.no gjennom testen i 2007, de klarte seg ganske godt:



Merk at slike automatiske tester selvsagt bør tas med en klype salt, de kan ikke oppdage alt, og endel av det de finner kan være såkalte falske positive.

Andre måter vi kan teste grad av universell utforming og brukskvalitet på, er å gå gjennom eksisterende sjekklister, som f.eks. leksjonen om brukskvalitet eller W3Cs retningslinjer for tilgjengelighet: <http://www.w3.org/TR/WCAG20/>

- ✓ **Testing av feilmeldinger.** 404 er et kjent begrep i websammenheng, og det er veldig vanlig å møte døde lenker, sider som er midlertidig nede eller rett og slett å skrive feil adresse. Her er resultatet av å gå til en side som var midlertidig nede på Tollpost Globe sine nettsider i oktober 2013:

www.tollpost.no/nb/E-tjenester/Sider/Miljøkalkulator.aspx


Min Side Logg Inn Om oss

TOLLPOST GLOBE
Alltid i forkant

PRIVAT E-TJENESTER LØSNINGER PRODUKTER KUNDESERVICE

Gå tilbake

404 - side eller fil ikke funnet



Vi beklager de ulemper dette har for deg.

Siden eller fil er ikke tilgjengelig. Nedenfor kan du se årsak og hvordan vi kan hjelpe deg.

Årsaken kan være følgende:

- Ønsket side er ikke lenger tilgjengelig.
- En feil har oppstått og siden du ønsket feilet.
- Siden oppdateres og vil være midlertidig utilgjengelig.
- Fant du denne siden via en link? Send beskjed til: support@tollpost.no så retter vi feilen.

Vi beklager dette, og håper du fant det du var på utkikk etter.

Søk med følgende stikkord for å finne relatert innhold:

Litt vanskelig å bli irritert på dem med en så hyggelig og hjelpsom 404-side.

- ✓ **Skjemavalidering.** Skjemaer er en veldig viktig del av nettsteder med brukerinteraksjon, og dermed blir skjemavalidering også veldig viktig. Hva skjer når brukerne dine skriver inn feil informasjon i et tekstfelt? Eller gjør andre feil?

Å gjøre skjemaer brukervennlige er et stort tema i seg selv. Her finner du lenker til et par artikler som du bør bruke noen minutter på med mindre du allerede er dreven på dette:

- ✓ Six Revisions: [10 Tips for Optimizing Web Form Submission Usability](#) (Raphael Caixeta, januar 2011).
- ✓ Smashing Magazine: [An Extensive Guide To Web Form Usability](#) (Justin Mifsud, november 2011).

Evaluering i praksis

I praksis kombinerer man gjerne flere typer evaluering for å få et best mulig sluttresultat. Her er et eksempel på hvordan man kan gå fram og utnytte de sterke sidene ved de forskjellige formene for evaluering i en utviklingsprosess:

1. Etter at de første designideene er på plass, kan man lage enkle prototyper og la brukere utforske dem i sitt naturlige miljø. Hva synes de? Bør noen av ideene forkastes umiddelbart, og hvilke bør vi satse på for videreutvikling? Bør noen av designene endres før vi jobber videre med dem? Enkle opportunistiske gjennomganger kan også brukes til å svare på konkrete spørsmål her (for eksempel om den konseptuelle modellen).

2. Lenger ute i prosessen, når vi har begynner å få på plass mer av designet, er tiden for klassiske brukertester i kontrollert miljø som fokuserer på kjernefunksjonaliteten til det vi lager. Da får vi forhåpentligvis luket ut alvorlige brukskvalitetsfeil, og noen gode råd å ta med videre i utviklingen. Dette er også et godt tidspunkt til å invitere eksperter på brukskvalitet (som ikke er utviklere i dette prosjektet og dermed ikke kjenner det altfor godt) til å gjøre en heuristisk gjennomgang.
3. Når produktet begynner å bli mer ferdig og det meste er på plass, kan det være en god idé å gjennomføre enda flere feltstudier – lån ut prototyper, og intervju brukere etter et par uker med produktet ditt. Hvilke justeringer bør gjøres før du sender produktet ut på markedet?

Oppsummering

Vi har nå sett på mange former for evaluering. Evaluering er et bredt begrep som blant annet inkluderer klassiske former for brukertesting, men også andre typer evalueringer som utviklere/interaksjonsdesignere kan gjøre på egenhånd. Evaluering er den fjerde fasen i IxD-syklusen, men vi kan gjøre enkle evalueringer på ethvert tidspunkt i en utvikling. Det er lett for en designer å se seg blind på det man lager, og enkle former for evaluering med jevne mellomrom er en god måte å passe på å holde seg på “rett spor”.

Til slutt noen spørsmål for å se hva du har fått med deg:

1. Hvorfor er evaluering av brukergrensesnitt viktig?
2. Forklar hvordan en klassisk (kontrollert) brukertest bør gjennomføres.
3. Forklar hva A/B-tester, feltstudier og opportunistiske evalueringer går ut på, og hva som er fordelene og ulempene med disse i forhold til klassiske brukertester.

Referanser

IAllenkelhet, bloggen til NetLife Research <http://www.iallenkelhet.no/>

Jakob Nielsens Alertbox på <http://www.useit.com/alertbox/>

Rogers, Yvonne; Sharp, Helen og Preece, Jenny. 2011.
Interaction Design: beyond human-computer interaction. Tredje utgave.
West Sussex, United Kingdom: John Wiley & Sons Ltd.
ISBN 978-0-470-66576-3

Toftøy-Andersen, Eli og Wold, Jon Gunnar. 2011.
Praktisk brukertesting. Første utgave.
Cappelen Damm AS.
ISBN 978-82-02-34350-7