

MMI AWESOMENESS

(__/)
\\(='.'=)/
('')_('')

Leksikalske hendelser:

Leksikalske hendelser formidles til viduet de intrefør i, og der definert ved:

- de minste enhetene i brukerinteraksjon
- direkte knyttet til de fysiske input-enhetene til datamaskinen

Leksikalske vs. syntaktiske hendelser

- Leksikalske hendelser er svært detaljerte:

- * mange enkelthendelser
- * komplisert logikk
- * kun noen overganger er relevante

- Syntaktiske hendelser:

- * tilsvarer en sekvens av leksikalske hendelser i en bestemt kontekst
- * abstraherer bort input-enheter og grafisk struktur
- * fokuserer på vesentlige kombinasjoner av betingelser og hendelser
- * nødvendigvis spesifikk for et/flere GUI -element(er)

BRUK SYNTAKTISKE DER MAN KAN

Brukskvaliteten er iflg ISO definert i forhold til bestemte brukere (brukergrupper) med bestemte oppgaver i bestemte omgivelser. Det som måles er anvendbarhet, effektivitet og tilfredstillelse.

- Brukergruppe: Ved å bare spørre de som har klart å bestille et produkt/betale gjør at man står igjen med en utvalg av brukere som man ikke klarte å inkludere, siden disse ikke kom seg videre i prosessen.

- Man har ikke muligheten til å si noe om hvor representative de som svarte på et spørreskjema var.

- Et spørreskjema vil omhandle ting som kanskje skjedde for flere uker siden

Når man har tatt i bruk et design? 3 forslag til å evaluere en brukskvalitet!

Brukbarhetstest

- Beskrivelse: En brukbarhetstest ville i dette tilfellet bety å la et representativt antall brukere utføre en del relevante oppgaver under kontrollerte forhold. Antall testpersoner er avhengig av hvor nøyaktig man ønsker resultatene skal bli. 5-8 personer gir erfaringsmessig et godt resultat.

-Fordeler: Gir kontroll på valg av brukere, oppgaver og omgivelser

- Ulemper: Krever godt kjennskap til målgruppen, oppgavene og bruksområdene.

Brukbarhetstesting gir god innsikt i problemer rundt selve brukergrensesnittets utforming og virkemåte, men det er en kunstig situasjon og man får ikke testet motivasjon, langtidseffekter og bruk i faktiske omgivelser.

Logging av bruk

- Beskrivelse: Mange programmer og web-servere har mulighet for å logge bruksmønstre. Så lenge dataene er anonyme så kan man bruke dette i en analyse av bruken. Dette gir innsikt i hvilke deler av programmet som brukes, hvor lenge brukerne sliter og hva de bruker det til.

-Fordeler: Automatisk logging kan gi data fra mange brukere, som igjen gir bedre pålitlighet på måldataene. Dette kan gi en pekepinn på hva av brukergrensesnittet som må endres på.

- Ulemper: Man får ingen kjennskap til faktiske brukere i forhold til alder, kjønn, motivasjon, forslag til forbedringer og hva de synes om tjenesten. Heller ikke hvordan de bruker tjenesten.

Nettbasert spørreskjema

- Beskrivelse: Det er relativt lett å lage et spørreskjema på nettet for å spørre brukerne om hva de synes om tjenesten, brukervennelighet +++

- Fordeler: Gir data fra mange brukere

- Ulemper : Det er vanskelig å vite hvor representativt utvalg som svarer på spørreskjemaet

Fokusgruppe

- Beskrivelse: Man samler en gruppe (4-8stk) representative brukere i et rom og presenterer netttjenesten. Gjennom tilbakemeldinger og diskusjoner får man feedback og innblikk i brukernes oppfatninger om tjenesten.

- Fordeler: En relativt enkel ting å gjennomføre ressursmessig. Gir mye feedback raskt og gir innblikk i brukergruppens preferanser og ønsker.

- Ulemper: Er dårlig på det konkrete om grensesnittets brukbarhet fordi brukerne ikke er nødt til å bruke systemet. Kan bli mye synsing som ikke er basert på faktisk bruks erfaring.

MVC arkitektur

- MVC er en programvarearkitektur for å skille data fra grensesnittelementer. Det kan sees på som en realisering av de to øverste lagene i en 3-lags arkitektur. MVC stammer fra Smalltalk, der man skilte mellom modellen, viewet og kontrolleren. Modellen inneholder alle data som forandrer seg under kjøring. Viewet tar seg av presentasjonen på skjermen og kontrolleren tar seg av input fra brukeren.
- Model, View og Controller objekter kobles under kjøring sammen slik at en modell kan ha flere view-controller par koblet til deg. Ved å automatisere oppdateringen av alle views når modellen endres, så innfører man abstraksjon som gjør det lettere å legge til nye views.
- Oppdatering av views gjøres ved at modellen holder en liste med view-controllere som er koblet til den, og når det skjer en endring på modellen så sendes det en beskjed til alle views at det nettopp har skjedd en endring. Hvert enkelt view vil da oppdatere seg selv ved å spørre modellen om hva som ble endret.

Affordance

- Fysisk: Gitt av kroppens- og objektets anatomi(natur) feks: hammeren, flasken
- Kulturelt: Konvensjoner som innlæres ("sånn åpner du en dør..")
- Gitt av konteksten/omgivelsene: (Ikke sånn, slik gjør man det på en iPhone..)

Constraints (Begrensninger)

- Visse ting kan man ikke gjøre med et produkt:
 - * Lego kan bare settes sammen på visse måter
 - * Visse knapper er grået ut og kan ikke trykkes på
- Fysisk: Gitt av produktets utforming
- Kulturelt: Visse ting gir mening, andre ikke
- Gitt av produktet/omgivelsene: Papir kan skrives på i en sammenheng, og brukes til å tenne bål i en annen

Feedback

- En handling skal ALLTID gi feedback
- * Når du trykker på en knapp skal det skje en handling..

Typer feedback

- Visuelle : muse-peker, modus-knapp, highlighting
- Auditive: HDD-lyd, kneppe lyd fra knapp

- Taktile: varierende motstand, forskjellig overflate
- feedback fullfører affordance, den erstatter ikke!**

Gestalt-prinsippene

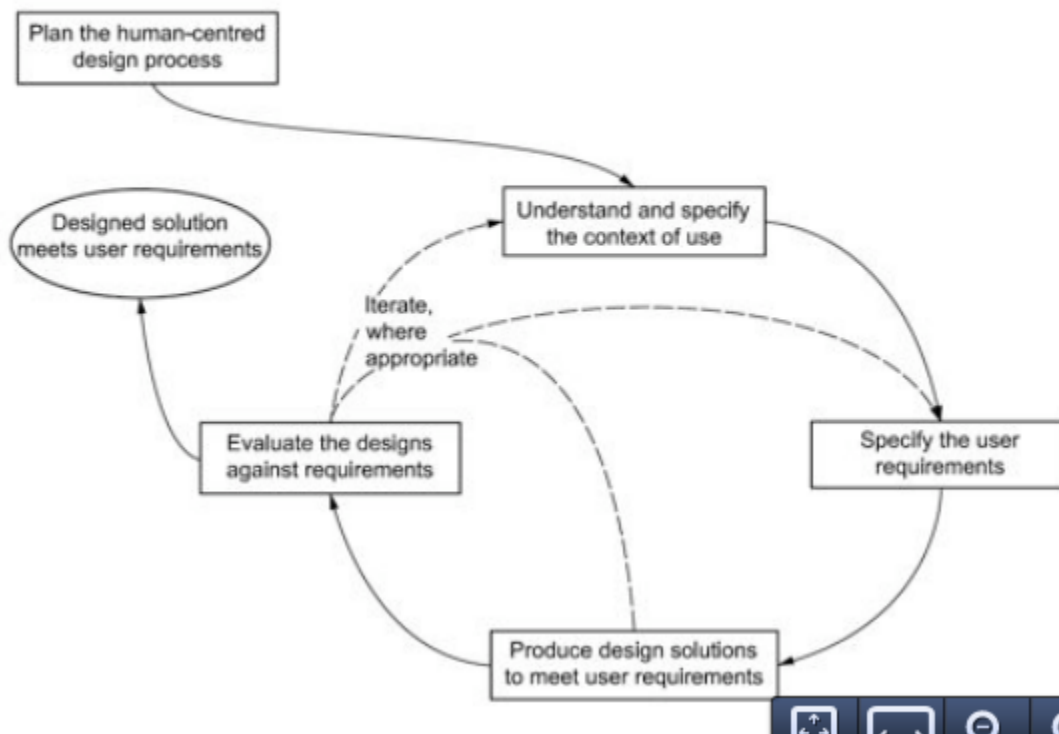
- Gruppering/nærhet
- Linje/kontinuitet
- Mental komplettering
- Likhet i form
- Likhet i farge
- Forgrunn/bakgrunn + reversering

WYSIWYG : "What you see is what you get"

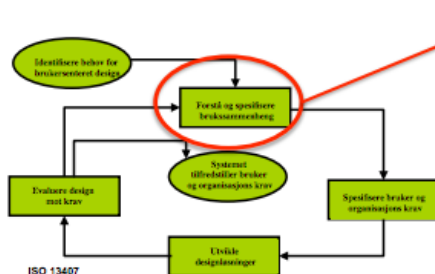
Tre modeller av et system

- Designer/utvikler: Konseptuell modell; - utviklers intensjon og spesifisering UML
- Bruker: Mental modell; - Det brukeren oppfatter, bekreftes/avkreftes
- System: Teknisk modell; - Implementasjon, UML 1-til-1 kode

ISO 9241-210: User-centred design



Fasene i bruker-sentrert design (1)



Forstå og spesifisere brukssammenheng

Teknikker for å forstå brukssammenheng:

- "Feltstudie": Observasjon og intervju.
- Fokusgrupper.
- Automatisk logging av bruksmønstre.

Teknikker for å formidle brukssammenheng:

- Personas basert observasjon og intervju.
- Scenarier av dagens situasjon basert på observasjon.
- Loggdata-analyser.

Fasene i bruker-sentrert design (2)



Spesifisere bruker og organisasjons krav

Teknikker for å spesifisere krav:

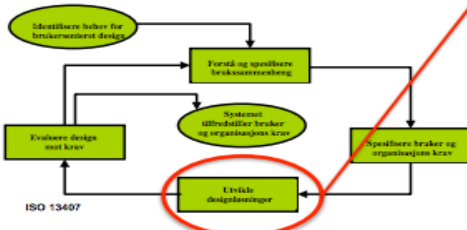
- Intervju med brukere.
- Designworkshop.
- Bruk av dramametoder.

Teknikker for å formidle krav:

- Kravlister
- Overordnede ikke-funksjonelle krav (bl.a. brukervennlighet)
- Scenarier og personas som viser tenkt system i bruk.



Fasene i bruker-sentrert design (3)



Utvikle designløsninger

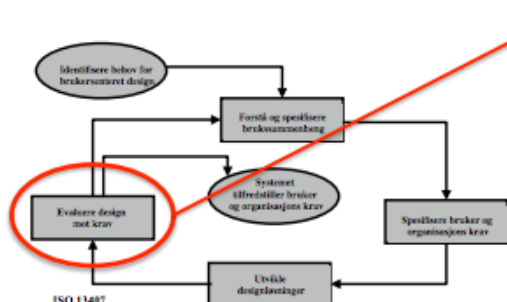
Teknikker for å utvikle designløsninger:

- Utvikle konseptuell modell / metafor.
- Prototyping:
 - Papirprototyper
 - Powerpoint, Flash el.l. prototyper
 - Funksjonelle prototyper
- Scenarier m/personas for løsning i bruk.

Teknikker for å formidle designløsninger:

- Framvising/utprøving av selve prototypene
- Vise scenarier m/personas for tenkt system i bruk.

Fasene i bruker-sentrert design (4)



Evaluere design mot krav

Teknikker for å evaluere designløsninger:

- Brukbarhetstesting.
- Fokusgrupper for feedback på løsning.
- Felt-tester og logging av bruk

Teknikker for å formidle resultat fra evaluering:

- Testrapport fra brukbarhetstest (ISO/CIF)
- Oppsummering av feedback fra fokusgruppe.
- Analyser av felt-tester og loggdata.

Definisjonen av brukskvalitet

def: "the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments"

- Anvendbarhet, effektivitet og tilfredsstillelse

- * for bestemte brukere
- * med bestemte mål
- * i bestemte omgivelser

Brukbarhetstester: Formativ vs. Summativ

Formativ:

- Evaluering med den hensikt å gi tilbakemelding for å forbedre et produkt
- Fokus på hvilke feil brukerne oppdager og hvordan dette kan meldes tilbake til utviklerne

Summativ:

- Evaluering med den hensikt å godkjenne eller måle brukskvaliteten på et produkt

- Fokus på målbare kriterier som oppgavegjennomføring, feilrate, tidsbruk og spørreskjema som subjektiv tilfredsstillelse

Arbeidsstegene i brukbarhetstestning:

1. Utvikling av målsettinger og hypotese for testen og utvikling av testplan
2. Skaffe brukere gjennom tilfeldig eller stratifisert utvalg
3. Forberede materiale og kontekst
4. Pilot-test
5. Velg forsøksleder
6. Utføre testen
7. Omforming av data til funn og anbefalinger

Ti punkter om å utføre brukbarhetstesting

1. Introduser deg selv
2. Beskriv hensikten med testen
3. Fortell deltakerene at de kan avbryte når de vil
4. Beskriv utstyret i rommet og begrensningene til prototypen
5. Lær bort hvordan man tenker høyt
6. Forklar at du ikke kan tilby hjelp under testen
7. Beskriv oppgaven og introduser produktet
8. Spør om det er noe de lurer på og kjør testen
9. Avslutt testen med å la brukeren uttale seg før du samler evt. løse tråder
10. Bruk resultatene

low-fidelity prototype

a prototype that is sketchy and incomplete, that has some characteristics of the target product but is otherwise simple, usually in order to quickly produce the prototype and test broad concepts.

high-fidelity prototype

a prototype that is quite close to the final product, with lots of detail and functionality. From a user testing point of view, a high-fidelity prototype is close enough to a final product to be able to examine usability questions in detail and make strong conclusions about how behavior will relate to use of the final product.

De 6 perspektivene på MMI som overordna metaforer for mennesket, datamaskinen og samspillet mellom dem

1. Systemperspektivet: Mennesket som datamaskin (UML use case)
2. Dialogperspektivet: Datamaskinen som menneske (utvikler må utforme dialog og intelligens)
3. Verktøyperspektivet: Mennesket som verktøybruker (Bruker datamaskinen som passivt verktøy)
4. Mediaperspektivet: Datamaskinen som medium (Bruker datamaskinen som et sted å lagre, samt presentere innhold)
5. Cyberspace: Maskinen forsvinner, den skaper verden (Rom-metafor: hvor, hva, hvem, virtual reality)
6. Tangible interaction: Informasjonen er i verden (Den fysiske verden vi kjenner blir forandret)

Brukervennlighet er snittet mellom tid-multipleksing (minst antall grensesnittelementer) og rom-multipleksing (færrest antall museklikk)

Design guidelines:

Jacob Nielsen's 10 punkter (heuristikker)

1. *Visibility of system status*
2. *Match between system and the real world*
3. *User control and freedom*
4. *Consistency and standards*
5. *Error prevention*
6. *Recognition rather than recall*
7. *Flexibility and efficiency of use*
8. *Aesthetic and minimalist design*
9. *Help users recognize, diagnose, and recover from errors*
10. *Help and documentation*

Don Normanns prinsipper

1. Visibility

Jo synligere funksjoner er, jo mer sannsynlig er det at brukeren vil skjønne hva han eller hun skal gjøre videre. Kontrasten er funksjoner man ikke kan se, disse er vanskelige å bruke, samt vanskelige å vite hvordan man skal bruke.

2. Feedback

Feedback handler om å sende tilbake informasjon om en handling som har blitt gjort og hva som har blitt fullført. Dette gjør at brukeren kan fortsette med aktiviteten. Flere forskjellige typer feedback er tilgjengelige for interaksjonsdesign, for eksempel lyd, risting, verbal eller kombinasjoner av disse.

3. Constraints

Designkonseptet som handler om begrense brukerinteraksjon på gitte tidspunkt. For eksempel så lar ikke musikkprogrammet Spotify deg søke etter musikk som ikke er lagret lokalt når maskinen ikke er koblet til nett.

4. Mapping

Mapping refererer forholdet mellom kontroller og deres effekt i den virkelige verden. Hvordan brukeren oppfatter et objekt i et grensesnitt er viktig for god design. Et godt eksempel på god, naturlig mapping er bruken av piltastene på et tastatur. Disse beveger kursoren opp og ned i dokument, og er naturlig å bruke, fordi vi vet hva piler betyr i den virkelige verden.

5. Consistency

Consistency referer til at design av grensesnitt bør bruke lignende operasjoner og lignende elementer for å utføre like oppgaver. Et konsistent grensesnitt følger regler, som at den bruker den samme operasjonen for å velge alle objekter. For eksempel kan et konsistent grensesnitt ha aktivert at ved å venstreklikke på et objekt, markeres det. Dette vil da gjelde for alle objekter i grensesnittet. Et grensesnitt som er inkonsistent, vil har unntak for regelen.

Affordance

Affordance er et begrep som brukes for å referere til et attributt av et objekt som lar brukere vite hvordan man bruker objektet. For eksempel inviterer en museknapp til å bli klikket ned, og den er fysisk begrenset av plastikkskallet sitt. På et veldig enkelt nivå, kan man si at "afford"

handler om å gi et hint om hvordan noe skal brukes. Hvis et fysisk objekt gir klare antydninger om hvordan det skal brukes, er det enkelt å bruke det.

Observasjonsteknikker - oversikt

Navn	Formål	Type data	God på	Dårlig på
<i>Spørreskjema</i>	Få svar på spesifikke spørsmål	I hovedsak kvantitative	Kan nå mange personer	Vi må vite hva vi skal spørre om
<i>Intervju</i>	Utforske. Få innblikk i et tema.	Kvalitativt	Gi innsikt i et problem. Dialog med bruker.	Tar tid, og viser ikke faktisk arbeidskontekst.
<i>Fokusgrupper og workshops</i>	Dialog med gruppe av brukere	Kvalitativt	Komme dypt i å forstå et problem. Brukerinvolvering	Ressurskrevende.
<i>Feltstudie</i>	Studie av faktisk brukskontekst	Kvalitativt	Forstå brukerens faktiske hverdag.	Tar tid. Krever ofte tillatelse, f.eks. på sykehus.
<i>Brukbarhets-testing.</i>	Evaluerer av prototyp eller produkt	Kvalitativt og kvantitativt	Evaluerer brukskvalitet av prototyp/produkt.	Krever at man har prototyp. Skjer ikke i faktisk kontekst.
<i>Logging av bruk</i>	Evaluerer av faktisk bruk	Kvantitativt	Faktisk bruk. Mange personer.	Gir ikke innsyn i årsaker til problemer.