

# MENNESKE-MASKIN- INTERAKSJON

Kompendium

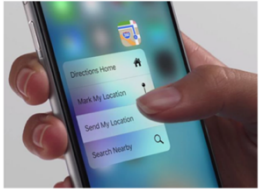
## SAMMENDRAG

Dette kompendiet er basert på forelesningsnotatene til emnet TDT4180 holdt ved NTNU våren 2019. Utfyllende informasjon er hentet fra boka «Interaction Design» av Preece, Rogers og Sharp og diverse artikler. Informasjon fra forelesningen er markert med (F), mens informasjon fra boka er markert med (B)

# F00 - Introduksjon

## Hva er interaksjonsdesign <sup>(F)</sup>

**Interaksjonsdesign (IxD)** er en fagdisiplin som har som oppgave å definere atferden til produkter og systemer som mennesker kan samhandle med. Mange produkter som krever brukerinteraksjon (eks: smartmobiler) har blitt utviklet med brukeren i tankene, og disse er generelt lette og fornøyeleg å bruke. Andre produkter har blitt laget for å primært utføre et sett av funksjoner (eks: alarmklokke) og disse kan fungere effektivt, men være komplisert å bruke. **Hovedmålet ved interaksjonsdesign er å redusere de negative aspektene og forsterke de positive.** Det handler om å utvikle interaktive produkter som er lett, effektiv og fornøyeleg å bruke fra brukerens perspektiv.



## Interaksjonsdesign <sup>(B)</sup>

Interaksjonsdesign er designing av interaktive produkter som støtter måten mennesker kommuniserer og interagerer i deres liv både i hverdagen og i arbeid. Det handler om å lage brukeropplevelser som forbedrer måten mennesker arbeider, kommuniserer og interagerer. Noen ulike definisjoner:

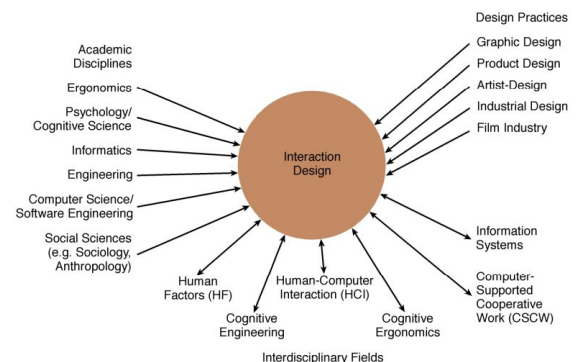
- «Design av rom der mennesker kommuniserer og interagerer»
- «Kunsten av å legge til rette for interaksjoner mellom mennesker vha produkter og tjenester»

Interaksjonsdesign er et paraplybegrep som kan omfatte programvaredesign, produktdesign, web design, osv. Fokuset ligger på hvordan man designer brukeropplevelser.

Interaksjonsdesign er ikke forbundet med en bestemt måte å designe, men fremmer heller bruken av en rekke metoder, teknikker og rammeverk. Det er forskjell på å designe et interaktivt produkt og lage programvaren for produktet. En analogi på interaksjons- og programvare design kan være bygging av hus. Da vil interaksjonsdesign være arkitekten som ser på hvordan menneskene som skal flytte inn vil bruke huset, mens programvaredesign er byggingeniøren som fokuserer på å realisere prosjektet.

## Komponenter i interaksjonsdesign

Interaksjonsdesign er fundamentalt for alle felt som ser på forskning og design av datamaskinsystemer for mennesker



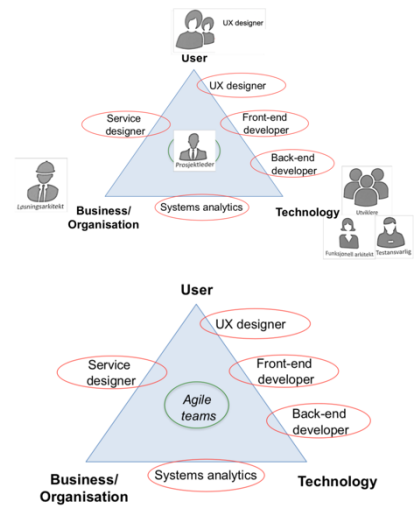
## Hva er en interaksjons/UX-designer

Designere må vite mye forskjellig om brukerne, teknologier og interaksjoner mellom dem for å kunne lage effektive brukeropplevelser. De må forstå hvordan mennesker reagerer på hendelser og hvordan de kommuniserer og interagerer med hverandre. De må også forstå hvordan emosjoner fungerer og hva som menes med estetikk, ønskelighet og fortellerens rolle i menneskelig opplevelser. Utviklere må også forstå forretningen, teknologien, produksjonen og markedsføringen. Det er vanskelig for en person å kunne alt dette, så derfor vil systemutvikling ha flere ulike roller (figur til høyre). **Interaksjonsdesign kommer innenfor gruppen**



**kalt UX designere**, og denne gruppen vil ofte bestå av mennesker med ulike bakgrunner for å fremme utvikling av nye kreative ideer. Flere mennesker involvert kan også gjøre prosessen mer komplisert, siden mennesker med ulike bakgrunner kan ha ulike perspektiv på hva som er viktig og kommunikasjon blir utfordrende.

På figuren til venstre kan vi se at UX-designere arbeider sammen med front-end og back-end utviklere, i grensesnittet mellom brukere og teknologi.



# F01 - Trender i MMI

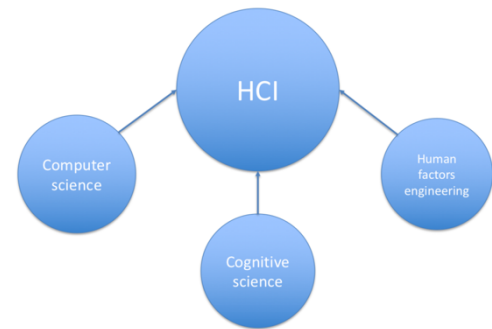
Læringsmål:

- Få kjennskap til ulike interaksjonsparadigmer
- Få kjennskap til noen sentrale visjoner og innovasjoner som har hatt stor betydning for hvordan vi interagerer med datamaskiner i dag.

## HCI (Human-Computer Interaction) og interaksjonsparadigmer (F)

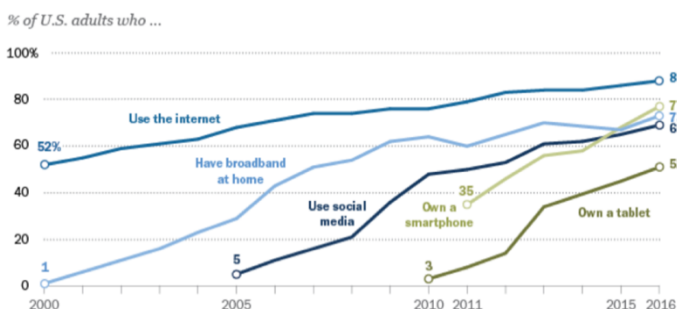
Human-Computer Interaction (HCI) ser på design og bruk av datateknologi, fokusert på grensesnittet mellom brukere og datamaskiner. Begrepet ble først tatt i bruk tidlig på 1980-tallet. HCI er basert på:

- **Datavitenskap** – studiet av prosesser som interagerer med data og som kan representeres som data i form av program.
- **Kognitiv vitenskap** – studiet av minnet og dens prosesser, med fokus på hvordan nervesystemet representerer, behandler og transformerer informasjon.
- **Menneskeligfaktor ingeniør (ergonomi)** – bruk av psykologiske og fysiske prinsipper i design av produkter, prosesser og system for å redusere menneskelige feil og øke produktivitet, sikkerhet og behag.

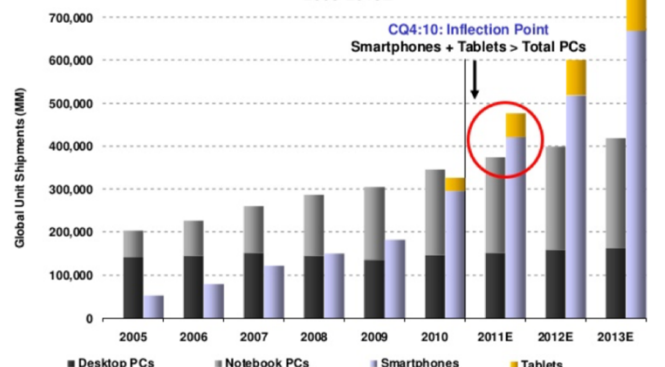


Figurene under viser hvordan anskaffelse og bruk av IKT har utviklet seg de siste årene. Som vi kan se har det vært en stor økning i andel mennesker som bruker Internett, smartmobiler, nettbrett, sosiale medier, osv.

The evolution of technology adoption and usage



Global Unit Shipments of Desktop PCs + Notebook PCs vs. Smartphones + Tablets 2005-2013E



Mainframe computing  
(First wave, ca. 1960-1985)Personal computing  
(Second wave, ca. 1985-2000)Ubiquitous computing  
(Third wave, ca. 2000-present)

## Interaksjonsparadigmer VIKTIG (F)

For å forstå HCI historien må vi forstå serien av paradigmeskifter.

Interaksjonsparadigmer beskriver hvordan forholdet mellom mennesker og datamaskiner har endret seg som følge av at ny datateknologi har bitt utviklet. **Disse paradigmene har ikke bare endret hva vi interagerer med (eks: PC, bærbar, mobil, osv.), men også hvor og i hvilke sammenhenger vi interagerer med datamaskiner.** Interaksjonsparadigmene beskriver hvordan **brukskonteksten har endret seg.** Tre viktige interaksjonsparadigmer, kalt *Three waves of computing* er:

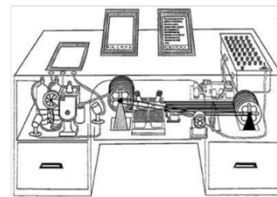
1. **Mainframe computing (first wave, ca. 1960-1985)** – stormaskiner krevde flere brukere av gangen og kunne ofte være på størrelsen med et rom.
2. **Personal computing (second wave, ca. 1985-2000)** – utviklingen av mikroprosessor gjorde at datamaskiner kunne bli mindre, kraftigere og billigere, og dermed mer egnet for personlig bruk.
3. **Ubiquitous computing (third wave, ca. 2000-present)** – utviklingen av internett, smartmobiler, nettbrett, smartklokker, osv. har gjort at datamaskiner kan brukes hvor og når som helst. Dette har ført til en allestedsnærværende databehandling.



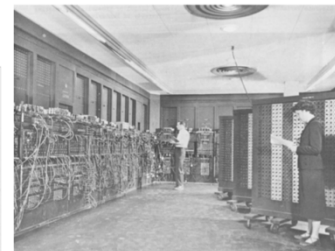
## Noen viktige milepæler i MMI VIKTIG (F)

Følgende er noen viktige milepæler i menneske-maskin-interaksjon historie:

- **1945-1959:** utviklingen av **ENIAC** som er verdens første programmerbare elektroniske datamaskin. Den kunne utføre 5000 operasjoner per sekund, noe som er en milliondel av det en laptop kan utføre i dag. Samtidig brukte den 1000 ganger så mye energi. I tillegg blir **Vision Memex** (mekanisk lagringsenhet) utviklet.
- **1960-tallet:** utviklingen av **TX2 Sketchpad** som er den første interaktive grafiske editoren. Den bruke en lypenn til å tegne på skjermen. Den første **datamusen** og **Dynabook** (personlig datamaskin til barn) blir laget.
- **1970-tallet:** utviklingen av **Alto computer** som var en personlig datamaskin som brukte datamus, grafisk brukergrensesnitt og nettverk (LAN). Det ble også utviklet objektorientert programmering (smalltalk) og nettverk (LAN).



MEMEX



ENIAC



Dynabook



Xerox Alto og Smalltalk



Datamusen



Xerox Star



Ubiquitous computing

- **1980-tallet:** utviklingen av vindussystemet **Xerox Star** som systematisk involverte brukerne i designprosessen. Det ble benyttet ikoner, mapper, menyer, knapper, dialogbokser, osv. I denne tiden var det en kommersialisering av PCen (økt salg).
- **1990-tallet:** utviklingen av Internett og mobiltelefonen fremmet visjonen av **ubiquitous computing** (dvs. allestedsnærværende databehandling). **Mark Weiser** sa «*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it*». ISO 9241-11 og ISO 13407 ble viktig (mer senere).
- **2000-nåtiden:** det har blitt utviklet **smarttelefoner med touch-grensesnitt** og utallige applikasjoner kan lastes ned via App store, google store, osv. I tillegg har sosiale medier som Facebook og Twitter fått en stor rolle i menneskers liv. Internet of things gjør at stadig nye enheter blir koblet opp mot internettet.

**Fokus på brukeren og brukeropplevelsen var lite systematisert før omtrent 1980** og den tidligste store industrielle satsningen var Xerox PARC i 1975. I 1980-1985 ble det etablert brukbarhetslaboratorier på IBM, HP, SUN og SONY. I dag har alle store bedrifter fokus på brukeropplevelsen, for eksempel Telenor, DnB, Statoil, osv. Alle store universitet tilbyr også MMI-fag, for eksempel Stanford, MIT, NTNU, osv.

# F10 - Brukskvalitet og brukeropplevelsen

Læringsmål:

- Forstå hva brukskvalitet (*usability*) er i hht. ISO 9241-11
- Forstå hva definisjonen betyr for design og brukerevaluering av produkter
- Få innsikt i relaterte konsepter som brukeraksept og brukeropplevelsen

## Godt og dårlig design <sup>(B)</sup>

Et sentralt fokusområde i interaksjonsdesign er å **utvikle interaktive produkter som er brukbare**, som vil si at de er lett å lære, effektive å bruke og gir gode brukeropplevelser. Ved å identifisere svakheter og styrker ved ulike interaktive produkter, kan vi begynne å forstå hva det betyr at noe er brukbart eller ikke. To eksempler på design:

- **Telefonsvarer**, to muligheter:

- **Dårlig design** – et hotellrom har en telefon der det trengs flere steg for å få og etterlate en melding, og instruksjonene gis på et ark og av en stemme i telefonen. Dette er et dårlig design fordi det er irriterende, forvirrende, ineffektivt (mange steg for grunnleggende oppgaver) og vanskelig å bruke. Det trengs flere steg for å se om man har fått noen meldinger og det er ikke åpenbart hva man skal gjøre siden informasjon blir delvis gitt av systemet og arket.
- **Godt design** – et alternativt design er klinkekule svarmaskinen (se figur), der klinkekuler brukes for å representere antall mottatte meldinger. Dette er et godt design fordi det bruker kjente fysiske objekter som visuelle indikatorer, det er estetisk gledelig, fornøyet å bruke, krever kun ett steg for å utføre kjernehandlinger, det er et enkelt, men elegant design og det gir mindre funksjonalitet. Designet benytter seg av brukerens kunnskap om den virkelige verden. Det er et eksempel på **interaktive produkter designet med brukeren i fokus**. Designet er ikke praktisk i hotellsammenheng, siden det ikke er robust (klinkekuler kan bli tatt) og det sjekker ikke hvem brukeren er. Det er **viktig å ta hensyn til hvor produktet skal brukes og av hvem!**



- Fjernkontroll, to muligheter:

- **Dårlig design** – mange fjernkontroller blir designet med mange rekker med små, flerfargede, navngitte knapper som ofte er vilkårlig plassert i forhold til hverandre. Det er vanskelig å finne riktig knapp, noe som er frustrerende.
- **Godt design** – TiVo kontrolleren har store knapper som er tydelig markert og logisk plassert, slik at de blir lettere å lokalisere og bruke sammen med meny grensesnittet på TV-en. Fjernkontrolleren ble designet med en form som passer i håndflaten. Det ble brukt fargerike knapper og ikoner som skiller seg ut og er derfor lett å identifisere i mørket. Denne kontrolleren er et resultat av en **bruker-sentrert designprosess**, der brukerne ble involvert i designet for å få tilbakemelding.



Godt design	Dårlig design
<ul style="list-style-type: none"> <li>- Estetisk gledelig</li> <li>- Fornøyet å bruke</li> <li>- Krever få steg</li> <li>- Enkelt, men elegant design</li> <li>- Mindre funksjonalitet</li> <li>- Benytter brukerens kunnskap om verden</li> <li>- Logisk plassering av knapper</li> <li>- Hører på hva brukerne ønsker</li> <li>- Tar hensyn til hva brukerne er gode og dårlige til</li> </ul>	<ul style="list-style-type: none"> <li>- Irriterende</li> <li>- Forvirrende</li> <li>- Ineffektivt (mye steg for lite funksjonalitet)</li> <li>- Vanskelig å bruke</li> <li>- Ikke åpenbart hva man skal gjøre</li> <li>- Spredt informasjon</li> <li>- Lytter ikke til brukerne</li> </ul>

I interaksjonsdesign ser man på hvordan man kan optimalisere brukernes interaksjon med et system, miljøet eller produktet, slik at det støtter og utvider brukernes aktiviteter på en effektiv, brukbar og nyttig måte.

## Brukskvaliteten VIKTIG (F)

For diamanter vil kutt, karat, farge og klarhet være et **evalueringskriterier for kvalitet**, mens for skihopping vil evalueringskriteriene være lengde og stil (svev og landing). Den engelske termen **usability** kan oversettes til **brukskvalitet, som er et sentralt begrep innenfor interaksjonsdesign**. For å evaluere brukskvaliteten blir det gjennomført såkalte **brukertester (usability test)**.



**Brukskvalitet** = i hvilken grad produktet er anvendbart, effektivt og skaper tilfredshet i bruk av bestemte brukere, med bestemte mål, i en bestemt brukskontekst. Det beskriver en relasjon mellom produktet og dets brukskontekst.

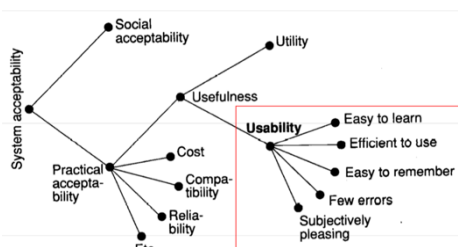
ISO 9241-11 (1998)

**Brukskonteksten** er brukere, oppgaver, utstyr (maskinvare, programvare og materialer) og det fysiske og sosiale miljøet der produktet brukes. **Brukerkonteksten er altså brukerne, oppgavene og omgivelsene, som er tre uavhengige variabler.**

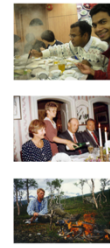
Følgende brukes for å måle brukskvaliteten til produktet i en bestemt brukskontekst:

1. **Anvendbarhet (effectiveness)** = hvor nøyaktig og komplett kan brukerne oppnå målene? Klarer brukeren å benytte designløsningen for å gjennomføre oppgavene sine og i så fall hvor godt og hvor mange oppgaver?
2. **Effektivitet (efficiency)** = ressursbruken for å oppnå målene med denne graden av nøyaktighet og kompletthet. Hvor mye tid trengs for å løse oppgaven gitt designløsningen?
3. **Tilfredshet (satisfaction)** = brukerens subjektive mening om hvor egnet designløsningen er. Hvordan oppleves og beskrives designløsningen av brukerne?

**Disse variablene vil avhenge av hverandre.**



Nielsen definerte god brukskvalitet til å bety at teknologien er lett å lære, effektiv å bruke, lett å huske, har få feil og er subjektivt fornøyet (se figur). Dette vil være avgjørende for at teknologien skal aksepteres av brukeren.

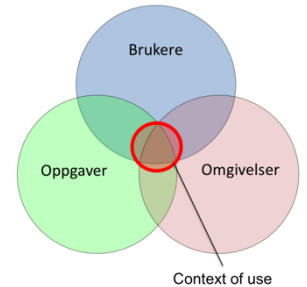


<b>OK</b>	<b>+/-</b>	<b>X</b>
<b>+/-</b>	<b>OK</b>	<b>X</b>
<b>X</b>	<b>+/-</b>	<b>OK</b>

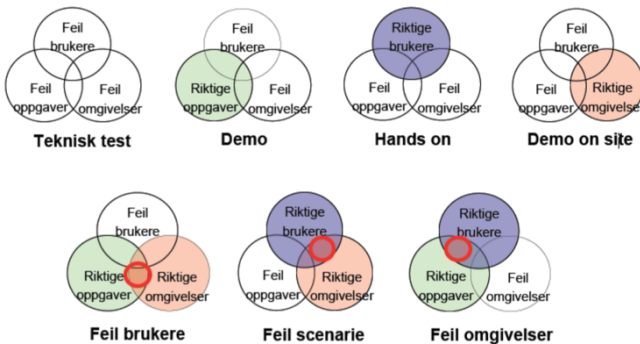
### Evaluering av brukskvalitet (F)

**Brukskvaliteten til et produkt vil være avhengig av brukskonteksten**, som vi kan se på **usability-matrisen** (se figur). Her er det tre produkter i tre ulike brukskontekster, og som vi kan se av matrisen vil **brukskvaliteten være en relasjon mellom produktet og brukskonteksten**. Det er bare meningsfylt å måle brukskvalitet når man kan svare på:

- **Hvem** er brukerne av produktet? = Brukere
- **Hva** ønsker de å bruke produktet til? = Oppgaver
- **Hvor** skal produktet brukes? = Omgivelser



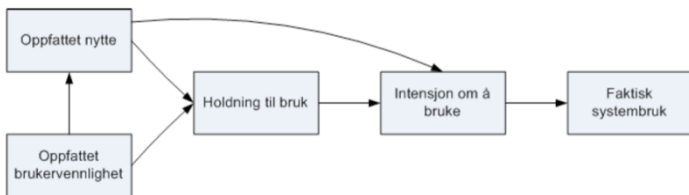
**Altså, må vi definere brukskonteksten (hvem, hva og hvor) for å kunne måle brukskvaliteten.** Hvis vi har feil brukskontekst vil vi få en svekket evaluering av målene på brukskvaliteten. For eksempel kan vi ha feil brukere, feil scenario eller feil omgivelser:



Når vi skal evaluere hvilken mobil som er best, vil det altså avhenge av brukssituasjonen (konteksten)

### Andre faktorer som påvirker brukerakseptansen (F)

**Det vil også være andre aspekter enn brukskvalitet som vil avgjøre om brukerne aksepterer et produkt.** For eksempel vil **nytteverdi være en avgjørende faktor for brukeraksept**. Et eksempel er boksåpneren som ikke alltid er enkel å bruke, men blir akseptert av brukeren fordi den har høy nytteverdi.

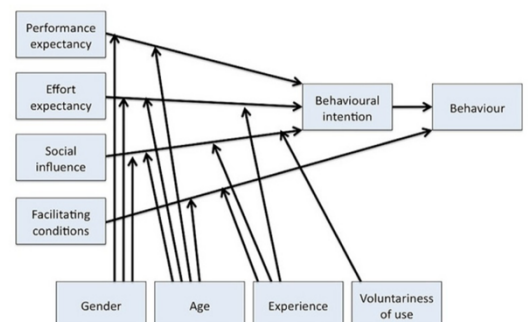


På figuren kan vi se at:

$$\text{Aksept} = \text{Nytteverdi} + \text{Brukskvalitet}$$

Dette kalles Teknologi Akseptansemodell (TAM)

Det vil være flere faktorer som spiller inn på brukerakseptansen, for eksempel kjønn, alder, erfaring, sosial påvirkning, osv. Disse vil avgjøre oppførselen til brukeren og dermed hvordan de aksepterer og tar i bruk teknologien. Dette kalles Unified Theory of Acceptance and Use of Technology (UTAUT).





## ISO 9241-11 - Brukskvalitet

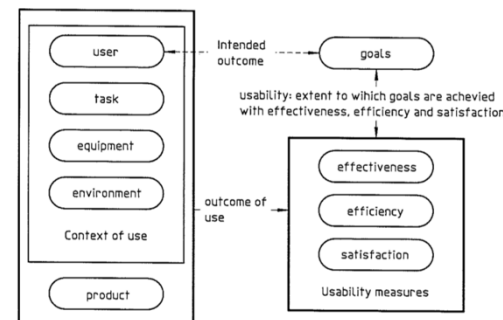
ISO 9241-11 definerer brukskvalitet (*usability*) og kan brukes for å spesifisere og evaluere brukskvaliteten til et produkt. Det forklarer fordelene ved å måle brukskvalitet mht. brukerprestasjonen og tilfredsheten, som måles etter hvilken grad målene ved bruken ble oppnådd, hvordan ressurser ble brukt for å oppnå målet og i hvilken grad bruken av produktet er akseptert. **Brukskvaliteten er avhengig av brukskonteksten og nivået til brukskvaliteten vil avhenge av situasjonen der produktet blir brukt.** Definisjoner:

- **Brukskvalitet** = i hvilken grad produktet kan brukes av spesifiserte brukere for å oppnå spesifiserte mål med anvendbarhet, effektivitet og tilfredshet i en spesifisert brukskontekst. Måles basert på:
  - **Anvendbarhet** = nøyaktigheten og komplettheten til oppnåelsen av målet
  - **Effektivitet** = ressursbruken ved oppnåelsen av målet
  - **Tilfredshet** = frihet fra ubehag og positive innstillinger mot bruken av produktet
- **Brukskontekst** = brukerne, oppgavene, utstyr og omgivelsene der produktet brukes
- **Oppgave** = aktiviteter som trengs for å oppnå et mål (fysiske eller kognitive)

Brukskvaliteten kan forbedres ved å legge til egenskaper som er til nytte for brukerne i en bestemt brukskontekst. ISO 9241-11 er et rammeverk som kan brukes for å identifisere aspekter ved brukskvaliteten og komponentene i brukskonteksten som må tas hensyn til for å evaluere brukskvaliteten til produktet. **Prestasjonen (anvendbarhet og effektivitet) og tilfredsheten til brukerne kan brukes for å måle brukskvaliteten til produktet i en bestemt brukskontekst.**

For å måle brukskvaliteten til et produkt, trenger man:

- **Beskrivelse av målene til produktet** – for eksempel for en telefonselger kan det være å håndtere kundebestillinger. Dette er et overordnet mål som kan deles inn i delmål.
- **Beskrivelse av brukskonteksten** – for ulike typer brukere kan kunnskap, egenskaper, utdanning, erfaring, osv. defineres. Oppgaver er aktivitetene som må gjennomføres for å oppnå et mål, og frekvensen og varigheten til oppgaver kan påvirke brukskvaliteten. Oppgaver beskrives med utgangspunkt i målet som skal oppnås. Relevante egenskaper ved utstyret og ved omgivelsene må beskrives. Beskrivelsen av omgivelsene kan involvere informasjon om nettverket, møbler, arbeidsplassen, temperatur, arbeidsmiljø, osv. Tabellene under gir eksempler på definisjon av brukskontekst.



Users	Tasks	Equipment
<b>User types</b> Primary Secondary and indirect users  <b>Skills and knowledge</b> Product skill/knowledge System skill/knowledge Task experience Organizational experience Level of training Input device skills Qualifications Language skills General knowledge  <b>Personal attributes</b> Age Gender Physical capabilities Physical limitations and disabilities Intellectual ability Attitude Motivation	Task breakdown Task name Task frequency of use Task duration Frequency of events Task flexibility Physical and mental demands Task dependencies Task output Risk resulting from error Safety critical demands	<b>Basic description</b> Product identification Product description Main application areas Major functions  <b>Specification</b> Hardware Software Materials Services Other Items

	Environment	
<b>Organizational environment</b>  <b>Structure</b> Hours of work Group working Job function Work practices Assistance Interruptions Management structure Communications structure  <b>Attitudes and culture</b> Policy on use of computers Organizational aims Industrial relations  <b>Job design</b> Job flexibility Performance monitoring Performance feedback Pacing Autonomy Discretion	<b>Technical environment</b> <b>Configuration</b> Hardware Software Reference materials	<b>Physical environment</b>  <b>Workplace conditions</b> Atmospheric conditions Auditory environment Thermal environment Visual environment Environmental instability  <b>Workplace design</b> Space and furniture User posture Location  <b>Workplace safety</b> Health hazards Protective clothing and equipment

Effectiveness measures	Efficiency measures	Satisfaction measures
Percentage of goals achieved;	Time to complete a task; Tasks completed per unit time;	Rating scale for satisfaction;
Percentage of users successfully completing task;	Monetary cost of performing the task	Frequency of discretionary use;
Average accuracy of completed tasks		Frequency of complaints

- **Beskrivelse av brukskvalitetsmål** – det må avgjøres hvordan anvendbarhet, effektivitet og tilfredshet skal måles. Dette valget vil avhenge av objektene som er involvert i målingen og brukskonteksten.
  - **Mål på anvendbarhet** = hvor nøyaktig og komplett målene blir oppnådd. Nøyaktighet er hvordan kvaliteten til output korresponderer med det spesifiserte kriteriet, mens kompletthet er hvor stor del av målmengden som har blitt oppnådd.
  - **Mål på effektivitet** = bruk av ressurser som mental eller fysisk innsats, tid, materialer eller finansielle kostnader for å oppnå målet. Det måles basert på forholdet mellom oppnådd anvendbarhet og ressursbruk
  - **Mål på tilfredshet** = hvor fri brukeren er fra ubehag og deres innstilling for bruk av produktet. Det kan måles etter antall positive og negative tilbakemeldinger. Målingen er subjektivt (eks: holdningsskala basert på spørreskjema) eller objektivt (eks: observasjon av kroppsspråk)

Man må være forsiktig med å generalisere resultatet til målingen av brukskvaliteten til andre kontekster som kan ha andre brukertyper, oppgaver eller miljø, siden det er **kontekstavhengig**.

Evalueringen av brukskvalitet blir brukt ved flere ulike tidspunkt i designprosessen:

1. **Spesifisering av brukskonteksten til produktet** – informasjon om brukernes egenskaper, deres mål og oppgaver og omgivelsene der oppgavene utføres, gir viktig informasjon som kan brukes i spesifiseringen av produktkravene
2. **Spesifisering av krav for brukskvaliteten** – ISO 9241-11 kan brukes for å spesifisere kravene til brukskvaliteten som produktet må tilfredsstille. Brukskonteksten må identifiseres og målene for anvendbarhet, effektivitet og tilfredshet må velges
3. **Produktutvikling** – definisjonen av brukskvalitet kan brukes for å etablere en felles forståelse av konseptet blant flere utviklerteam. Den kan også brukes for å ta objektive avgjørelser om det trengs endringer i designet for å øke brukskvaliteten.
4. **Spesifisering av produktegenskaper** – beskrivelsen av brukskonteksten kan brukes for å vurdere behovet for bestemte produktegenskaper
5. **Måling av brukskvalitet** – ISO 9241-11 gir informasjon som kan benyttes i brukertesten som måler brukskvaliteten. For eksempel kan beskrivelsen av brukerne benyttes for å avgjøre hvilke brukere som skal delta i evalueringen av produktet. Identifisering av mål brukes for å velge passende oppgaver og beskrivelsen av omgivelsene brukes i simuleringen for å sikre et gyldig testresultat.

Et produkt kan ha flere mål som bidrar til brukskvaliteten, for eksempel hvor lett det er å lære (se tabell).

Målinger på brukskvaliteten bør være basert på data som gir resultatet av at brukere interagerer med produktet. Dataen kan være objektiv eller subjektiv. Gyldigheten til brukertesten vil avhenge av hvor godt brukerkonteksten representerer den virkelige situasjonen der produktet skal brukes og egenskapene til målingen. Man kan ta målinger i «feltet» der man bruker reelle situasjoner for å evaluere brukskvaliteten eller man kan ta målinger på laboratoriet der konteksten blir kunstig skapt. Målinger på laboratoriet gir bedre kontroll på variabler og mer presise målinger, men den kunstige settingen kan gi urealistiske resultater.

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Meets needs of trained users	Number of power tasks performed; Percentage of relevant functions used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Meets needs to walk up and use	Percentage of tasks completed successfully on first attempt	Time taken on first attempt <sup>1</sup> ; Relative efficiency on first attempt	Rate of voluntary use
Meets needs for infrequent or intermittent use	Percentage of tasks completed successfully after a specified period of non-use	Time spent re-learning functions <sup>1</sup> ; Number of persistent errors	Frequency of reuse
Minimization of support requirements	Number of references to documentation; Number of calls to support; Number of accesses to help	Productive time <sup>1</sup> ; Time to learn to criterion <sup>1</sup>	Rating scale for satisfaction with support facilities
Learnability	Number of functions learned; Percentage of users who manage to learn to criterion	Time to learn to criterion <sup>1</sup> ; Time to re-learn to criterion <sup>1</sup> ; Relative efficiency while learning	Rating scale for ease of learning

## Eksempel

Målet er at VidiPro skal brukes for å ringe eller ta i mot anrop og dele visuell informasjon på et kontor

- **Oppgave:** Kommuniser med stemme og dele visuell informasjon i et kontormiljø over en utvidet periode
- **Brukskontekst:** Brukerne har brukt VidiPro for å dele visuell informasjon minst 60 ganger over en periode på minst en måned
- **Anvendbarhet:** Nøyaktighet: Mindre enn 2% av alle anrop resulterer i kobling til feil nummer. Kompletthet: Minst 95% av anropene fullføres suksessfullt
- **Effektivitet:** Gjennomsnittlig oppsett av videoanrop er ikke mer enn 30 sekunder
- **Tilfredshet:** Minst 90% av brukerne foretrekker VidiPro over alternativt utstyr

## Brukeropplevelsen VIKTIG (F)

**Brukeropplevelsen (UX = user experience) er hvordan produktet oppfører seg og brukes av mennesker i den virkelige verden.** Det er mange aspekter som definerer brukeropplevelsen, for eksempel brukskvaliteten, nytteverdien, utseende/estetikk, identitet, forventninger, erfaringer, pris, støttetjenester, kultur, osv.



**Brukeropplevelse (UX)** = brukerens opplevelse under og etter faktisk bruk og/eller forventninger knyttet til bruk. Det er samlingen av brukerens følelser, tro, preferanser, opplevelser, fysisk og psykisk respons og oppførsel i sammenheng med et produkt, et system eller en tjeneste **før, under og etter** bruk. Brukeropplevelsen er **totalinntrykket av produktet**

ISO 9241-210 (2010)

**Brukeropplevelsen kan altså begynne før bruken,** for eksempel når man går inn på Apple store og får en følelse av eksklusivitet og spenning. Opplevelsen vil påvirkes av blant annet merkevarebygging, presentasjon, funksjonalitet, interaktiv oppførsel, produktøkologi, støttetjenester, tidligere erfaringer, evner, personlighet, holdninger og brukskontekst.



## Brukeropplevelse (B)

**Brukeropplevelsen (UX = user experience) er hvordan produktet oppfører seg og brukes av mennesker i den virkelige verden.** Det omfatter alle aspektene ved brukerens interaksjon med selskapet og dets tjenester og produkter. Alle produkter som brukes av noen vil ha en brukeropplevelse, for eksempel aviser, ketchup flasker, osv. **Brukeropplevelsen er hva mennesker føler om produktet og deres tilfredshet og glede når de bruker det, ser på det, holder det, åpner det og lukker det.** Det sier noe om deres totale inntrykk om hvor godt det er å bruke. En brukeropplevelse kan ikke designes, men man kan designe trekk som gir en brukeropplevelse (eks: telefondeksel kan designes til å være glatt og fast i håndflaten for å gi en tilfredsstillende brukeropplevelse).

Brukeropplevelsen er totalopplevelsen av et produkt. **For brukere vil brukergrensesnittet være systemet,** siden de forholder seg til det konkrete istedenfor abstraksjoner og dette vet designere. En mobiltelefon er ikke en datamaskin med radio, taster, skjerm og batteri. Det er en mobiltelefon, en Android-telefon eller en iPhone-telefon, og den er min. **Brukeraksept krever derfor at brukergrensesnittet blir akseptert.** Et godt brukergrensesnitt er en nødvendig, men ikke tilstrekkelig betingelse for kvaliteten til totalsystemet.

Det er mange aspekter ved brukeropplevelsen, for eksempel brukskvaliteten, innholdet funksjonaliteten, det estetiske, utseende, følelsen og sensuell og emosjonell appell. Andre faktorer er helse, sosial kapital, kulturell identitet, alder, etnisitet, rase, familiestatus, osv. Det finnes flere rammeverk og retningslinjer for hvordan disse faktorene skal brukes i interaksjonsdesign. For eksempel er det et rammeverk som ser på brukeropplevelsen basert på hvordan den føles av brukerne. Dette bestemmes av:

- **Det sensuelle** – hvor absorbert brukerne er i deres interaksjon
- **Det emosjonelle** – hvordan følelser er flettet sammen med opplevelsen
- **Det kompositoriske** – hvordan indre tanker til brukeren gir mening til opplevelsen
- **Det romlig-timelige** – tid og rom der opplevelsen tar sted.

Disse trådene kan brukes av designere for å tydeligere se forholdet mellom teknologi og opplevelser. Man kan se på hele opplevelsen, istedenfor fragmenter (eks: brukskvaliteten).

Evaluering av det som har blitt laget er hjertet til interaksjonsdesign, og det blir ofte gjort vha en brukersentrert tilnærming til design. Det er viktig å forstå brukskonteksten, fordi det som fungerer for en brukergruppe kan være fullstendig unyttig for en annen («*one size does not fit all*»). Når man lærer mer om mennesker kan det avsløre ukorrekte antagelser om ulike brukergrupper, for eksempel at eldre ønsker at ting er store (studier har vist at de har ingen problemer med å interagere med informasjon med standard størrelse). Det er også viktig å være klar over kulturelle ulikheter, for eksempel måling av tid i USA og Europa. Ulike farger, bilder og strukturering av informasjon kan brukes for å appellere til mennesker i ulike land.

For å forstå brukerne må designere ha et klart hovedmål ved utviklingen av et interaktivt produkt for brukerne. Målene bør klassifiseres mht. brukskvalitet og brukeropplevelse. Disse har ikke et klart skille, siden de er gjensidig avhengig.

#### Mål for brukskvalitet <sup>(B)</sup>

Brukskvalitet innebærer å sikre at interaktive produkter er lette å lære, effektive å bruke og fornøylig fra brukerens perspektiv. Interaksjonen må optimaliseres, og målene blir ofte formulert som detaljerte spørsmål innenfor:

- **Anvendbarhet** – hvor godt produktet gjør det som det er ment å gjøre, for eksempel «Er produktet i stand til å la brukerne kjøpe det de ønsker?»
- **Effektivitet** – hvordan produktet støtter brukerne når de utfører deres oppgaver, for eksempel «Når brukerne har lært å sjekke tapte anrop, hvor mange steg krever det?»
- **Sikkerhet** – hvordan brukerne beskyttes fra farlige forhold og uønskede situasjoner, for eksempel «Hvilke feil kan brukerne gjøre og hvordan kan de komme seg etter feilen?»
- **Nytteverdi** – hvordan produktet gir riktig type funksjonalitet, slik at brukere kan gjøre det de ønsker, for eksempel «Gir produktet funksjonalitet som lar brukerne utføre deres oppgaver slik de ønsker?»
- **Lærbarhet** – hvor lett systemet er å lære, for eksempel «Kan brukerne lære hvordan de bruker produktet vha utforskning av grensesnittet?»
- **Minneverdighet** – hvor lett det er å huske hvordan man bruker produktet, for eksempel «Hvilken type grensesnittstøtte gis for å hjelpe brukerne å huske hvordan de bruker produktet?». Dette er viktig for interaktive produkter som sjeldent brukes.

Disse spørsmålene brukes for å lage brukskvalitetskrav som er bestemte målsettinger som brukes for å evaluere hvordan produktet kan forbedres mht. brukerens prestasjon. Vanlige

brukskvalitetskrav er tiden som trengs for å utføre en oppgave (effektivitet), for å lære en oppgave (lærbarhet) og antall feil som utføres når man utfører en oppgave (minneverdighet).

#### Mål for brukeropplevelsen <sup>(B)</sup>

Tabellen gir en rekke mål ved brukeropplevelsen. Flere av disse er subjektive mengder om hvordan systemet føles for en bruker. **Brukeropplevelsen handler om brukerens opplevelse fra deres perspektiv, mens brukskvaliteten er mer objektiv og handler om hvor nyttig og produktivt systemet er fra systemets perspektiv.** Prosessen av å velge riktig begrep for å best beskrive brukerens følelser når de interagerer med et produkt ved en gitt tid og sted kan hjelpe designere til å forstå de mange sidene og den endrende naturen til brukeropplevelsen.

Ønskede aspekter	Uønskede aspekter
- Tilfredsstillende	- Kjedelig
- Fornøyeelig	- Frustrerende
- Engasjerende	- Gir skyldfølelse
- Nyttig	- Irriterende
- Spennende	- Barnslig
- Underholdende	- Fordømmende
- Hjelpende	- Ikke-fornøyeelig
- Motiverende	
- Utfordrende	
- Gøyalt	
- Overraskende	
- Belønnende	

# F11 – Intro til brukersentrert design

Læringsmål:

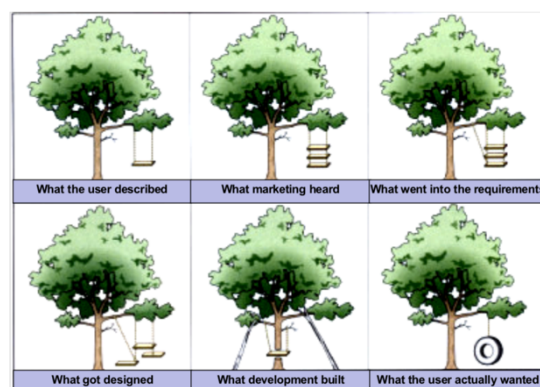
- Få en grunnleggende innsikt i den brukersentrerte designprosessen, mht.:
  - Motivasjon og bakgrunn
  - Kjennetegn
  - Stegene i prosessen

## Brukersentrert design <sup>(F)</sup>

Brukersentrert design er en designfilosofi som gir fremgangsmåte for å designe produkter med høy brukskvalitet. Det bygger på kunnskap om brukerne og brukerkonteksten, og involvering av brukere i designprosessen står sentralt. Det kan variere hvor mye brukerne blir involvert:

- Sporadisk involvering = brukerrepresentanter brukes som testsubjekter
- Kontinuerlig involvering = brukerrepresentanter blir en del av prosjektteamet

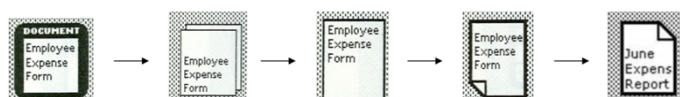
Designmetoder og teknikker innenfor brukersentrert design involverer intervju, spørreskjema, bruk av designguidelines (heuristikk), personas og scenarier, workshops, prototyping, feltobservasjoner, brukertesting (*usability testing*) og logging av faktisk bruk. Figuren illustrerer hvorfor det er viktig å involvere brukerne. Markedet, designere, utviklere og brukere vil ofte ha ulike tanker om hva som forventes av produktet, og det er viktig at brukernes tanker blir lagt vekt på for å bedre brukskvaliteten og brukeropplevelsen.



## Motivasjon og bakgrunn <sup>VIKTIG (F)</sup>

Opprinnelsen til brukersentrert design var utviklingen av vindussystemet Xerox Star (ca. 1980) som systematisk involverte brukerne i designprosessen. Det ble benyttet skrivebord, ikoner, mapper, menyer, vinduer, knapper, dialogbokser, objektorientert arkitektur, osv.

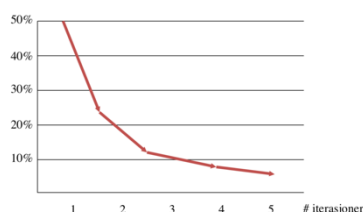
Brukergrensesnittet var basert på resultatet fra en rekke brukertester av ulike versjoner. Designmetoden var en gradvis forfining. For eksempel kan vi se på utviklingen av ikonet for dokumenter på skrivebordet, som blir gradvis finere:



En slik gradvis forfining bygger på et **iterativt design**, der man designer en tjeneste og utfører en brukertest for å identifisere nødvendige endringer i designet, og deretter gjentas prosessen. I



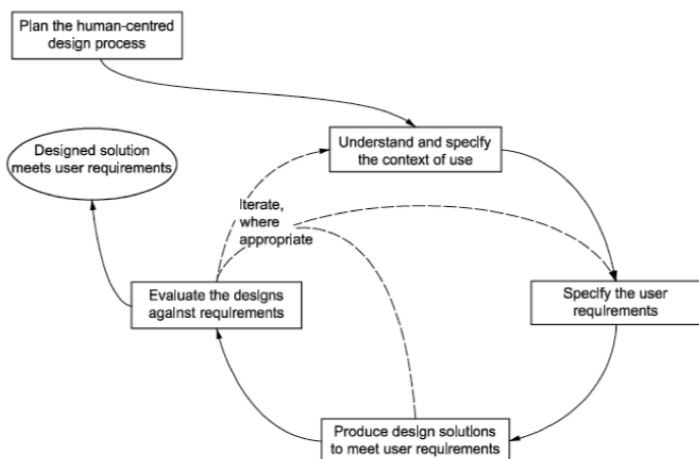
tidlig MMI ble det foreslått at dersom teorien var på plass vil man kunne lage generelle retningslinjer som fører til brukervennlige produkter, men dette er feil. **Det er umulig å få til riktig design med en gang, så derfor må man planlegge en iterativ prosess.** I begynnelsen av designet vil feilraten være høy, men **vha flere iterative steg med redesign og brukertesting vil feilraten reduseres (se figur)**



## ISO 9241-210 (2010) (F)

### Følgende er fasene ved brukersentrert design:

1. **Fase 1: Forstå brukskonteksten** – identifiser hvem som hovedsaklig kommer til å bruke produktet, hvorfor de vil bruke produktet, hva de krever av produktet og i hvilke sammenhenger de vil benytte seg av produktet.
  - **Teknikker for å spesifisere brukskonteksten** = feltstudier, intervjuer, gruppeintervju, automatisk logging av bruksmønstre, rollespill eller litteraturstudie.
  - **Teknikker for å formidle brukskontekst:** personas basert på observasjon og intervju, scenarier av dagens situasjon basert på observasjon eller loggdata-analyser.
2. **Fase 2: Spesifisere brukerkrav** – når vi vet hva og under hvilken kontekst produktet skal brukes kan vi spesifisere krav til produktet.
  - **Teknikker for å spesifisere krav:** intervju med brukere og andre interessenter, fokusgrupper eller rollespill
  - **Teknikker for å formidle krav:** kravlister, overordnet ikke-funksjonelle krav (blant annet brukervennlighet) eller scenarier og personas som viser tenkt system i bruk. Tabellen ned til høyre viser et eksempel på en brukerkravliste.
3. **Utvikle designløsninger** – når vi vet hva vi skal lage kan vi avhengig av hvor langt vi er kommet i designprosessen begynne å produsere et produkt.
  - **Teknikker for å utvikle designløsninger:** prototyping brukes for å lage ikke-funksjonelle (papirprototyper) eller funksjonelle (kjørende prototyper) løsninger. Alternativt kan scenarier med personas brukes for å illustrere løsningen i bruk eller man kan bruke designretningslinjer (eks: Norman)
  - **Teknikker for å formidle designløsninger:** utprøving/demo av selve prototypen eller visning av scenarier med personas
4. **Evaluere design mot krav** – etter å ha utviklet en designløsning må man få brukerens tilbakemelding av produktet
  - **Teknikker for å evaluere designløsninger:** brukbarhetstester (brukertester) i laboratoriet, fokusgrupper for feedback på løsning, felttester og logging av bruk eller skjema (SUS).
  - **Teknikker for å formidle resultat fra evaluering:** testrapport fra brukertest, oppsummering av feedback fra fokusgruppen eller analyse av felttester og loggdata



Fasene ved brukersentrert design

ID	Krav	Må	Bør	Kan	Kommentarer/forklaring	Kriterier
1.	Applikasjonen må gi brukeren mulighet til å rapportere sin subjektive oppfattelse av dagsform kontinuerlig og på en entydig måte.	1				
2.	Applikasjonen må gi brukeren en kvittering (verifikasjon) hver gang rapporter om hans/hennes dagsform mottas av Helsevakta	1			Viktig at tidspunkt for siste rapport er lett tilgjengelig vi brukergrensesnittet. Historikk kan fungere som referansepunkter	
3.	Applikasjonen bør gi brukeren en oversikt over tidligere innrapportert data om dagsform.		2			
4.	Applikasjonen bør tilby brukeren mulighet for å få automatiske varsler (påminnelser) dersom han/hun ikke har rapportert dagsform i løpet av en dag.		2			
5.	Applikasjonen må gi brukeren mulighet for å be om å bli kontaktet av relevant helsepersonell via: - Telefon - Videotelefon		1 3			Uavhengig av rapportert dagsform. Kvittering
6.	Applikasjonen bør gi brukeren tilgang på relevant informasjon i tekst og videoforamt. Dette inkluderer informasjon om KOLS, helsefare knyttet til røyking (og hjelp til å slutte), fysisk aktivitet og kosthold.		2			Tilgjengelig uten pålogging.
7.	Applikasjonen kan med fordel tilby brukeren mulighet til å registrere og rapportere når han/hun er på en medisinsk (type medisiner, oppstart, avslutning)			3		Kvittering
8.	Applikasjonen bør kunne gi brukeren en oversikt over tidligere registrerte medisinkurer han/hun har gjennomgått.			3		
9.	Applikasjonen bør tilby brukeren tilgang til individuell plan.			3		
10.	Applikasjonen bør tilby brukeren tilgang til behandlingsplan.			3		
11.	Applikasjonen må kunne gi mulighet for å at brukeren kan logge ut ved aktiv handling eller automatisk etter inaktivitet i fem minutter.		1			
12.	Applikasjonen må være i tråd med DIFs retningslinjer for universell utforming med hensyn til fargebruk, tekststørrelse og kontrast.					For å øke mulighet for bruk til tross for Kort oppsummering av viktigste punkter, legg til lenke
13.	Applikasjonen må ha høy brukskvalitet.	1			Lenke til SUS beskrivelse, norsk versjon i appendix. Når skal det måles, på hvor mange	SUS score > 75

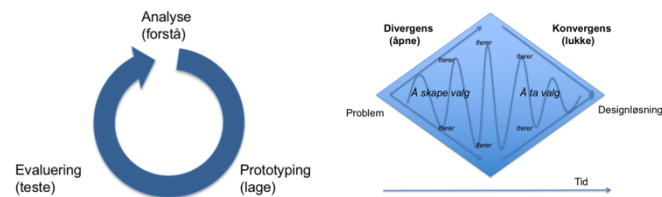
Eksempel på brukerkravliste

Legg merke til at det er en iterativ prosess!

Kjennetegn ved den brukersentrerte designprosessen:

- **En iterativ prosess!**
- **Aktiv involvering av brukere**
- Klar forståelse av brukerne, deres oppgaver og omgivelsene de befinner seg i
- Tverrfaglige utviklingsteam
- Sterkt fokus på at utvikler må forstå og kunne formulere brukerkonteksten
- **Et design som er kontinuerlig forbedret gjennom brukersentrert evalueringer.**

Det iterative designet vil være en syklus som består av **analyse (forståelse) prototyping (lage) og evaluering (teste)**. Input er et problem, mens output er en designløsning som utvikles over tid. Den iterative prosessen kan deles inn i divergens der problemet åpnes og iterasjonene blir større, og konvergens der iterasjonene blir mindre og resultatet er en designløsning.



## ISO 9241-210 – Brukersentrert design

ISO 9241-210 beskriver hvordan man går frem for å lage produkter med høy brukskvalitet. Brukersentrert design er en tilnærming i utviklingen av interaktive systemer der man forsøker å lage brukbare og nyttige systemer ved å fokusere på brukerne, deres behov og krav. Denne tilnærmingen styrker anvendbarheten og effektiviteten og forbedrer menneskelig velvære, bruker tilfredshet, tilgjengelighet og bærekraft. Definisjoner:

- **Brukersentrert design** = tilnærming til systemdesign og utvikling der målet er å gjøre interaktive systemer mer brukbare ved å fokusere på bruken av systemet og benytte menneskelige faktorer og kunnskap om brukskvalitet
- **Prototype** = en representasjon av hele eller deler av et interaktivt system, som brukes for analyse, design og evaluering. Det kan være alt fra en papirprototype til et fullt funksjonelt interaktivt system.
- **Brukeropplevelse** = brukerens oppfatning og respons som resultat av bruken og/eller forventet bruk av et produkt, system eller tjeneste. Det inkluderer brukerens følelser, tro, preferanser, oppfatninger, responser og oppførsel som skjer før, under og etter bruk. Brukeropplevelsen er en konsekvens av varemerket, presentasjon, funksjonalitet, ytelse, interaktiv oppførsel, hjelpemidler, osv.  
**Brukskvalitet kan brukes for å vurdere deler av brukeropplevelsen**
- **Brukergrensesnitt** = alle komponentene i et interaktivt system som gir informasjon og kontroll til brukeren slik at brukeren kan utføre bestemte oppgaver

Noen av fordelene ved brukersentrert design:

- Det lager brukbare produkter som ofte er mer suksessfulle, både teknisk og kommersielt
- Det reduserer støttekostnaden siden brukere kan forstå og bruke produktene uten hjelp
- Det reduserer risikoene for skade på helsen til brukerne og øke sikkerheten
- Det forbedrer brukeropplevelsen og brukskvaliteten
- Det kan forbedre varemerket og dermed gi et konkurransefortrinn
- Det gir bedre identifikasjon av funksjonskrav
- Det øker sannsynligheten for å fullføre prosjektet innen gitt tid og budsjett



## Kjennetegn ved brukersentrert perspektiv VIKTIG

Brukersentrert perspektiv kan brukes med mange ulike design- og utviklingsprosesser, men uansett hvilken prosess som brukes må brukersentrert tilnærming følge prinsippene:

1. **Designet er basert på en forståelse av brukerne, oppgavene og omgivelsene** – designet av et produkt bør ta hensyn til menneskene som skal bruke dem og andre interessenter. Alle relevante brukere og interessenter bør derfor identifiseres. En hovedkilde til systemfeil er å lage systemet basert på en ufullstendig forståelse av brukerne. Informasjon om brukskonteksten er essensiell input til designprosessen.
2. **Brukere blir involvert i design og utviklingen** – involvering av brukere er en verdifull kilde for informasjon om brukskonteksten, oppgavene og hvordan brukere sannsynligvis vil arbeide med det fremtidige produktet. Brukerinvolveringen bør være aktiv, enten ved at de deltar i designet, fungerer som kilde for relevant data eller evaluerer løsninger. Brukerne som involveres bør ha egenskaper og erfaring som reflekterer de endelige brukerne av produktet. Hvordan brukerne involveres kan variere avhengig av type prosjekt og hvor langt i utviklingen prosjektet er.
3. **Designet er drevet av brukersentrert evaluering** – evaluering av design med brukere og forbedring basert på tilbakemeldinger brukes for å redusere risikoen av å lage et produkt som ikke tilfredsstillende behovene til brukere eller organisasjonen. Designløsninger testes mot scenarioer i den «virkelige verden» og resultatet brukes for å gradvis forfine løsningen. Det brukes også i den endelige akseptansen av produktet
4. **Prosesen er iterativ** – det mest passende designet kan som regel ikke oppnås uten iterasjon, der en sekvens av steg gjentas til ønsket utfall oppnås. Iterasjon brukes for å gradvis eliminere usikkerhet i utviklingen av interaktive system. Beskrivelser, spesifikasjoner og prototyper blir forfinet når ny informasjon oppnås for å redusere risikoen for at systemet ikke tilfredsstillende brukerkravene. Menneske-maskin interaksjonen er så kompleks at det er umulig å spesifisere alle detaljene ved begynnelsen av utviklingen. Mange behov vil dukke opp under utviklingen ettersom brukerne blir bedre til å uttrykke behovene i respons til potensielle løsninger.
5. **Designet tar hensyn til hele brukeropplevelsen** – designet må ta hensyn til ulike faktorer som påvirker brukeropplevelsen, for eksempel støtte, produktpakking, merkevare, markedsføring, osv. Når aktivitetene utført av brukerne og teknologien blir spesifisert må man ta hensyn til brukernes styrker, begrensninger, preferanser og forventninger.
6. **Designteamet inkluderer multidisiplinære egenskaper og perspektiv** – brukersentrert designteam trenger ikke å være store, men det må være tilstrekkelig mangfoldig. Det bør inkluderes medlemmer som har kunnskap innenfor menneskelige faktorer, brukskvalitet, menneske-maskin interaksjon, markedsføring, merkevare, brukergrensesnitt, forretningsanalyse, programvareutvikling, osv. Prosjekt vil tjene på å ha medlemmer som tilsammen har et stort ferdighetsgrunnlag.

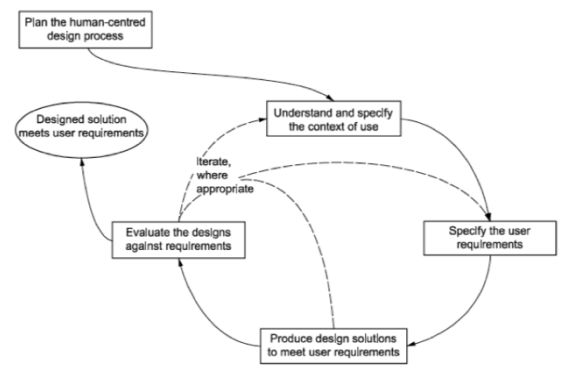
## Planlegging av brukersentrert design

Aktiviteter ved brukersentrert design bør starte ved begynnelsen av prosjektet som en del av prosessen for å formulere initialt konsept. Deretter vil designet fortsette gjennom hele prosjektet. Planen for brukersentrert design bør følge samme disipliner som prosjektet for å sikre at den følges.

## Stegene i brukersentrert design VIKTIG

Utfordringer ved design av interaktive system er at det kan være flere brukergrupper og andre interessenter med ulike behov, brukskonteksten kan variere mellom brukergruppene, noen krav vil dukke opp underveis i utviklingen, brukerkrav kan være omfattende og motstridende til hverandre, osv.

Brukersentrert design håndterer disse utfordringene vha fire steg (se figur):



1. **Spesifisering av brukerkonteksten** – egenskapene til brukerne, oppgavene og omgivelsene definerer konteksten der systemet skal brukes. Det er nyttig å samle og analysere informasjon om nåværende kontekst for å forstå og deretter spesifisere konteksten som vil gjelde for det fremtidige systemet. Resultatet er en beskrivelse av brukskonteksten som skal inkludere brukergrupper og interessenter, egenskaper til brukerne, målene og oppgavene til brukerne og tekniske, sosiale og fysiske omgivelser til systemet.
2. **Spesifisering av brukerkravene** – behovet til brukere og andre interessenter må identifiseres, mens man tar hensyn til brukskonteksten. Dette gir hva brukerne skal oppnå og inkluderer begrensninger som settes av brukskonteksten. Brukerkravene gir grunnlaget for design og evaluering av interaktive systemer.
3. **Produsering av designløsninger** – brukersentrert design har som mål å oppnå god brukeropplevelse ved å ta hensyn til den gjennom hele designprosessen. Designløsninger blir produsert basert på beskrivelsen av brukskonteksten, resultatet til evalueringer, standarder for design og brukskvalitet og kunnskapen til designteamet. Oppgaver og grensesnitt blir designet for å møte brukerkravene, samtidig som man tar hensyn til den totale brukeropplevelsen (tilfredshet, anvendbarhet og effektivitet). Scenarioer, simuleringer, modeller og andre prototyper lar designere kommunisere designet til brukere og interessenter for å få tilbakemelding. Dette gjør det enklere for teammedlemmer å kommunisere med hverandre og brukere, flere alternativer kan evalueres, tilbakemelding kan inkorporeres tidligere i designet og kvaliteten forbedres. I tidlige faser bør enkle prototyper brukes for å utforske alternative designløsninger. Kompliserte prototyper kan føre til motstand i endring av designet siden mye tid og penger er brukt på prototypen.
4. **Evaluering av design** – evaluering basert på brukerens perspektiv er et krav for brukersentrert design. Selv i tidlige fase skal designet evalueres for å oppnå en bedre forståelse av brukernes behov. Brukersentrert evaluering er ikke praktisk ved alle steg, så i stedet kan man bruke modellering og simulering (brukere deltar ikke direkte). Brukersentrert evaluering brukes for å samle mer informasjon om brukernes behov, gi tilbakemelding om styrker og svakheter ved designløsningen fra brukerens perspektiv, vurdere om brukerkravene har blitt oppnådd og etablere grunnlag for sammenligning av design. Evalueringen gjennomføres som brukertester, felttester, fokusgrupper for tilbakemelding på løsning, osv.

# F20 – Designkonvensjoner og GUI-komponenter

Læringsmål:

- Forstå hva som menes med konvensjoner i designsammenheng og spesielt mht. GUI-kontrollelementer
- Forstå konvensjonens betydning for et produkts brukskvalitet
- Få innsikt i ulike grupper av kontrollelementer, og designkonvensjonene tilknyttet enkelte kontrollelementer

## GUI-kontrollelementer <sup>(F)</sup>

Xerox star grensesnittet førte til utviklingen av WIMP og deretter GUI (*Graphical User Interface*), som ga nye muligheter for hvordan brukere kan interagere med et system og hvordan informasjon kan presenteres på grensesnittet. Det ble mulig å visuelt designe grensesnittet på nye måter, og designet inkluderer bruken av farger, typografi og bilder. Utfordringen i dag er å designe GUIs som passer best for grensesnitt på nettbrett, smarttelefoner og smartklokker. I stedet for å bruke mus og tastatur, vil standarden for de fleste brukerne være å sveipe og touche med en enkel finger.

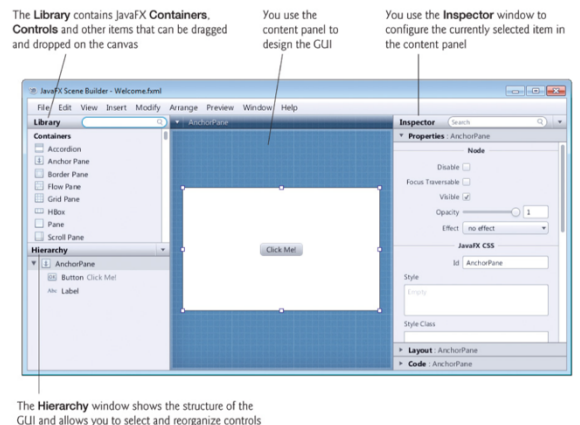
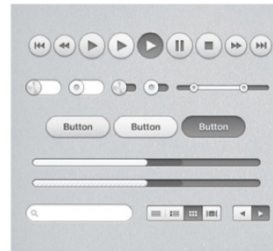
**GUI-kontrollelementer er byggeklossene til GUI-designere**, og de kalles kontrollelementer, GUI-elementer, komponenter, widgets, osv. Mange av disse har sitt opphav i utviklingen av Xerox Star på 1980-tallet. I dag har de i stor grad blitt standardisert, altså ser de ofte like ut og tilbyr lik oppførsel på tvers av plattformer. **Det er gjennom kontrollelementene at brukeren får interagere med applikasjonen**, for eksempel kan de trykke på knapper eller gi informasjon i tekstfelt.

**Et brukergrensesnitt bygges opp ved å kombinere ulike typer kontrollelementer**, for eksempel tekstfelter, knapper, sjekkbokser, osv. En GUI-designer kan bruke toolkits for å gjøre dette, for eksempel SceneBuilder i JavaFX (se figur).

## Hva et kontrollelement består av

**Alle kontrollelementer har en klasse, en superklasse, egenskaper og dataverdi(er) og responderer på hendelser**, for eksempel museklikk eller når musepekeren holdes over et element. Et eksempel er sjekkboks som har:

- **Klasse** = `javafx.scene.control.CheckBox`
- **Superklasse** = `javafx.scene.control.ButtonBase`
- **Dataverdi(er)** = statusen som er avkrysset (checked), ikke avkrysset (unchecked) eller ikke gjeldende
- **Egenskaper** = høyde, vidde, tekstjustering, enabled/disabled, osv.
- **Hendelser** = for eksempel `ValueChangeEvent` som aktiveres når verdien i sjekkboksen endres



### Class CheckBox

```
java.lang.Object
  javafx.scene.Node
    javafx.scene.Parent
      javafx.scene.layout.Region
        javafx.scene.control.Control
          javafx.scene.control.Labeled
            javafx.scene.control.ButtonBase
              javafx.scene.control.CheckBox
```

## Grupper av kontrollelementer VIKTIG

GUI-kontrollelementene kan deles inn i følgende grupper:

- **Inputelementer** – sjekkbokser, radioknapper, brytere, knapper, tilstandsknapper, rullegardin-meny, listebokser, tekstfelt og datafelt
- **Navigasjonselementer** – sliders, ikoner, sidenavigatører (*pagination*) og scroll bærer
- **Informasjonselementer** – dialogbokser, merkelapper, verktøytips, ikoner, progressbærer, notifikasjoner og meldingsbokser
- **Containers** – faner, paneler og accordian (sammenslåbar liste)

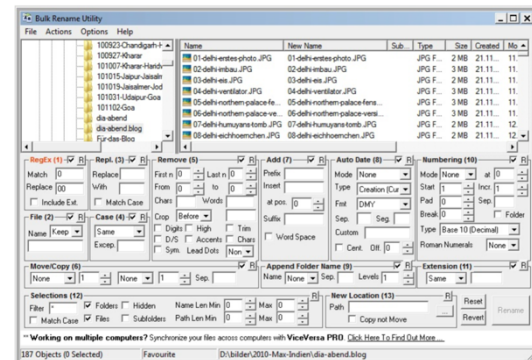
**Mange GUI-kontrollelementer er metaforer, altså har de motstykker i den virkelige verden.** For eksempel vil kontrollpaneler inneholde knapper, mens papirskjema kan inneholde tekstfelt, sjekkbokser, ikoner, osv.



## Kontrollelementer og brukskvalitet

GUI-kontrollelementer er ment å være **gjenkjennbare for brukeren**, altså vil de skape en forventning til hvordan systemet skal oppføre seg. **Brukskvaliteten til et GUI er i stor grad avhengig av at designeren følger konvensjonene som er knyttet til de ulike GUI-kontrollelementene.**

Kontrollelementene i seg selv vil altså ikke garantere høy brukskvalitet, fordi dette krever at de brukes på «riktig» måte. Dette er illustrert på figuren der kontrollelementene gir dårlig brukskvalitet, siden grensesnittet er uoversiktlig, vanskelig å lære og vil ikke være tilfredsstillende for brukeren.



## Konvensjoner VIKTIG

**Konvensjonene er et sett med alminnelige anerkjente retningslinjer eller normer.** Det er altså spilleregler eller begrensninger som oppfordrer til visse typer handlinger og motvirker andre. **Konvensjoner er kulturelt betinget eller tillært og de utvikler seg over tid.** Bruken av konvensjoner forutsetter et praksisfellesskap. Det tar lang tid å etablere konvensjoner, men det er vanskelig å ikke bruke de når de først er etablert.

Kontrollelementene som brukes for å lage grafiske brukergrensesnitt (eks: knapper, menyer, sjekkbokser, osv.) har egne konvensjoner. De ulike elementene er altså ment å bli brukt på bestemte måter. **Det er viktig å følge konvensjonen for bruk av kontrollelementer, fordi det vil avgjøre brukskvaliteten til brukergrensesnittet. Brukeren har forventninger til hvordan hen kan interagere med systemet og dersom konvensjonen til GUI-komponentene ikke er fulgt vil ikke brukeren forstå systemets oppførsel.** Dette vil være frustrerende for brukeren og dermed vil det redusere brukerens tilfredshet (= reduserer brukskvaliteten). GUI-designere bør derfor ha en veldig god grunn for å ikke følge konvensjonene. **Det er også viktig å være konsistens, altså sikre at oppførselen er lik i alle deler av grensesnittet og bruke like termer på ting som betyr det samme. I design av grensesnitt vil «less is more» gjelde, altså må man vurdere nøye om det er behov for GUI-komponentene i grensesnittet.** Det er fortsatt viktig med **brukertesting av designet**, selv om man følger konvensjonene som er tilknyttet elementene

Vi skal se på konvensjonen til noen GUI-kontrollelementer, og skiller mellom DOs og DON'Ts. Vi vil fokusere på DOs, altså konvensjonen vi skal bruke.

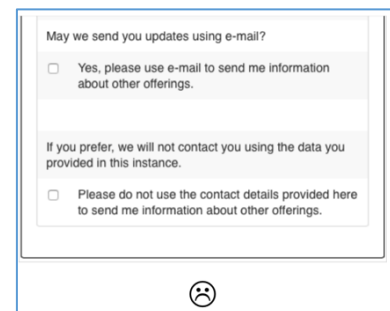
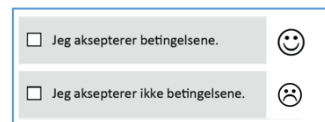
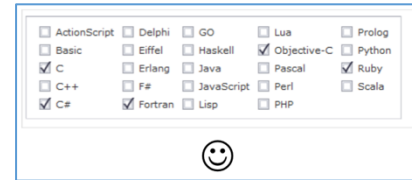
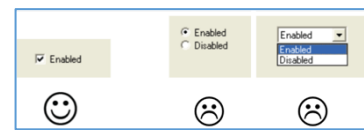
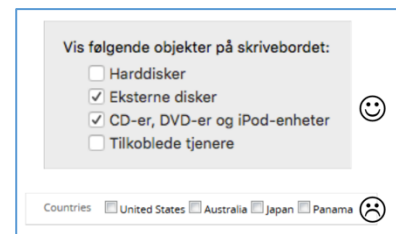
DOs and DON'Ts



## Sjekkboкс – inputelement

**Sjekkboксer brukes når brukeren kan gjøre flere valg (0...N) fra et sett med muligheter.** DOs for sjekkboксer:

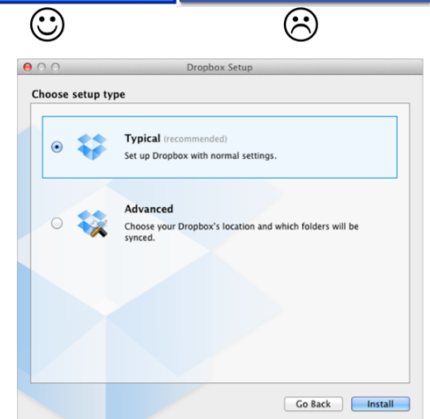
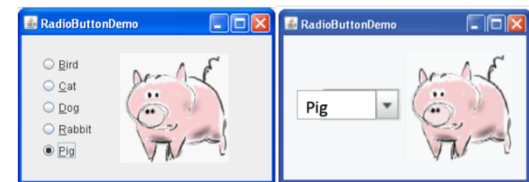
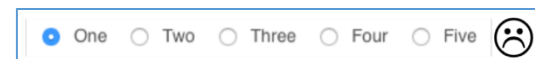
1. **Presenter valgene som en vertikal liste**, for da blir det lettere å sammenligne alternativene.
2. **Bruk én sjekkboкс når brukeren må velge én av to alternativer**, fordi det forenkler dialogen og tar mindre skjermplass.
3. **Bruk flere kolonner dersom listen er langt**, for da unngår man scrolling (alternativene blir mer synlig) og det blir lettere å sammenligne. Gruppering eller sortering vil hjelpe.
4. **Bruk positive og aktivt ladde beskrivelser knyttet til en sjekkboкс.** Det er viktig at brukeren enkelt kan forstå konsekvensen av valgene sine
5. **Sørg for at sjekkboксer ikke er gjensidig utelukkende.** På figuren kan vi se at brukeren kan gjøre gjensidig utelukkende valg ved å merke begge sjekkboксene. I dette tilfellet ville ett spørsmål med tilhørende sjekkboкс vært nok



## Radioknapper – inputelement

**Radioknapper brukes når brukeren kun kan velge ett alternativ blant flere gjensidig utelukkende alternativer.** DOs for radioknapper:

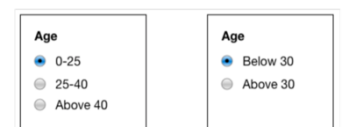
1. **Bruk vertikal liste når det er mange valgmuligheter** (kan bruke horisontal når det er få valg), fordi dette gir bedre oversikt og gjør det lettere å sammenligne valg
2. **Bruk radioknapper hvis det er få alternativer å velge mellom**, fordi det gjør at alle alternativene er synlig, slik at det blir lettere å sammenligne alternativene og brukerinteraksjonen blir mer effektiv. Alternativet er rullegardin-meny (brukes ved flere valgmuligheter)
3. **Hvis et alternativ er vanligere å velge eller anbefalt, sett dette som forhåndsvalgt**, for å redusere behovet for peking og klikking. For eksempel hvis de fleste brukerne velger «Pig» kan dette settes som forhåndsvalgt. **Forhåndsvalgte alternativer kan lede uerfarne brukere til å gjøre trygge valg**, for eksempel under installasjon og sikkerhetsinnstillinger. Dette kan videre forsterkes med å legge til «anbefalt» som merkelapp.



## Sjekkboксer og radioknapper

DOs som gjelder sjekkboксer og radioknapper:

1. **Sørg for at alternativene ikke overlapper hverandre og at hele mulighetsrommet er dekket**, slik at det blir lett å velge.
2. **Presenter valgmuligheter som logisk hører sammen i én gruppe.** For eksempel bør «Husdyr» og «Ville dyr» plasseres i hver sin gruppe. **Lag tydelige skiller mellom ulike grupper** i form av «luft».
3. **Sørg for at brukere må bekrefte valgene med en OK-knapp før de får effekt.** Dette er ikke nødvendig hvis valgene har liten konsekvens og lett kan rettes opp



Hva hvis du er 25? Hva hvis du er 30?

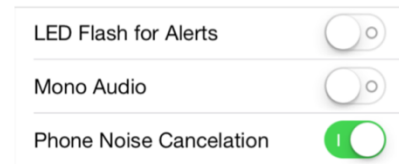
## Brytere – inputelement

Brytere blir ofte brukt i stedet for tradisjonelle sjekkbokser i mobile grensesnitt for å skape en forventning om at endringen skjer umiddelbart (lik oppførsel som lysbryter).

## Knapper – inputelement

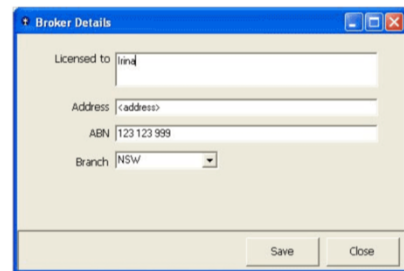
Knapper brukes når brukeren skal kunne aktivere en bestemt aksjon ved å trykke på knappen. DOs for knapper:

1. **Bruk ramme/kant og dybde**, slik at knappen blir enkel å kjenne igjen
2. **En knapp skal *alltid* ha samme aksjon**; en lagre-knapp er og blir en lagre-knapp
3. **Bruk entydig tekst og/eller symbol for å tydeliggjør for brukeren hvilken aksjon knappen fører til**, slik at brukeren vet hva som skjer når de trykker på knappen. På figuren er det uklart hva som er konsekvensen av å trykke Save og Close. Et forslag til ny løsning kan være OK (lukker og lagrer data), Cancel (lukker uten lagring av data) og Apply (lagrer data uten å lukke)
4. **Gjør knapper tilgjengelige (enabled) først når de har en logisk funksjon**. For eksempel på figuren bør knappen Close kun være synlig dersom ingen data er fylt inn.



OK

Arkiver

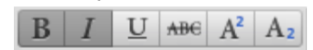


## Tilstandsknapper (toggles) – inputelement

Tilstandsknapper brukes når en bruker kan velge flere alternativer, der noen alternativer er gjensidig utelukkende (i motsetning til sjekkbokser). For eksempel ved tekstformatering i Word vil alternativene  $A^2$  og  $A_2$  være gjensidig utelukkende.

Tilstandsknapper brukes når alternativene kan fremstilles grafisk, altså uten tekst. DOs:

1. **Bruk symboler som er gjenkjennbare**, for eksempel understrek: U
2. **Bruk lett forståelige symboler for å representere tilstandene**. Symbolbruket er ofte standardisert, altså like symboler brukes på tvers av plattformer.

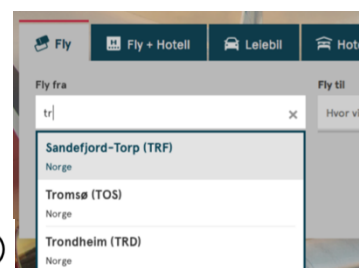
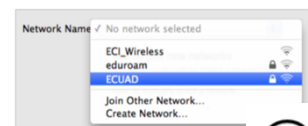


Når en tilstandsknapp trykkes vil **tilstandsendringen skje umiddelbart**, mens for radioknapper og sjekkbokser skjer endringen som regel først etter brukeren har bekreftet valget ved å klikke på en OK-knapp. **Tilstandsknapper bør derfor utløse handlinger med lav konsekvens som det er lett å gjøre om.**

## Rullegardin-meny (dropdown lists) – inputelementer

Rullegardin-meny brukes når brukeren kan velge ett alternativ blant flere mulige. Dette er likt radioknapper, men forskjellen er at det er **flere alternativer** og **antall alternativer er dynamisk**, for eksempel valg av trådløst nettverk. Det er for mange alternativer ( $> 7 \pm 2$ ) til at det er hensiktsmessig med plass til radioknapper, for eksempel valg av destinasjon i forbindelse med bestilling av flybillett. DOs:

1. **Tydeliggjør hvilket alternativ som er valgt**
2. **Pass på hva en nedtrykket rullegardin-meny dekker over**, siden brukeren ofte ikke kan se alle valgmulighetene (forutsetter scrolling) og menyen kan inneholde informasjon som er relevant for valget til brukeren
3. **Bruk autofylling/filtrering**, for å gjøre det lettere for brukeren å finne riktig alternativ



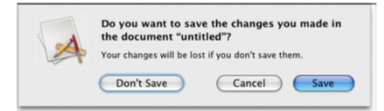
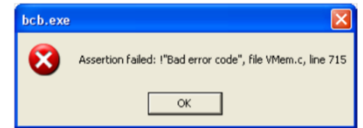
4. **Bruk maks to nivåer**, siden flere nivå krever mer fin-motorikk og brukeren må gjenta mange steg ved feil trykk



## Dialogbokser - informasjonselementer

Dialogbokser brukes for å gi viktige tilbakemeldinger som brukeren må ta stilling til. DOs:

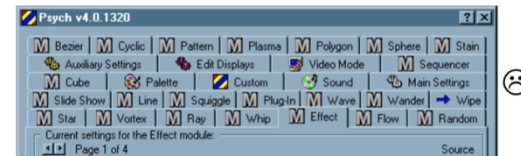
1. **Gi hva som har skjedd/vil skje**
2. **Gi hva brukeren kan gjøre og konsekvensen av dette**



## Faner (tab controls) – Containers

Faner brukes for å la brukeren skifte mellom ulikt skjerminnhold i en gitt kontekst, når det er viktig at brukeren hele tiden må se en bestemt del av brukergrensesnittet.

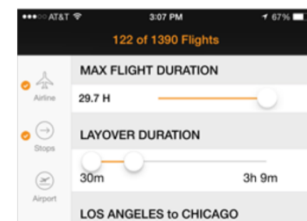
1. **Tenk gjennom hvilken fane som bør vises som standard**
2. **Sørg for at informasjon og aksjoner knyttet til samme fane logisk hører sammen**, slik at det blir lettere for brukeren å finne ønsket data
3. **Legg fanene på toppen av panelet for PC og bunnen av panelet for mobile enheter**
4. **Bruk  $7 \pm 2$  faner**, siden det kan være vanskelig for brukeren å huske innholdet på flere nivåer.



## Sliders – navigasjonselementer

Sliders brukes for å sette og vise en verdi innenfor et intervall med øvre og nedre grenseverdi. For eksempel kan det brukes for å stille inn lydvolument eller rangere et produkt. Det er uegnet for å gjøre et valg innenfor et mulighetsrom uten øvre og nedre grenseverdi, for eksempel i «fugl, katt, hund, kanin og gris». DOs:

1. **Bruk anslagsvise verdier**, siden det er vanskelig å sette spesifikk verdi hvis intervallet har høy granularitet og krav til nøyaktighet. Dette krever finmotorikk og er spesielt vanskelig i touch-grensesnitt.



Hvis høy granularitet (dvs. mange verdier) er nødvendig, bør man vurdere andre kontrollelementer, som for eksempel inntasting i tekstfelt.

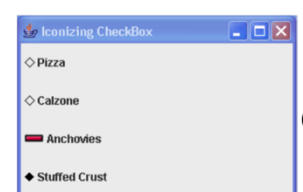
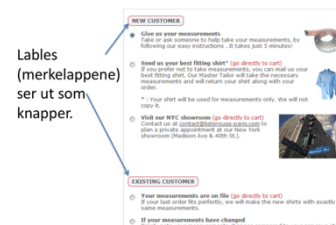
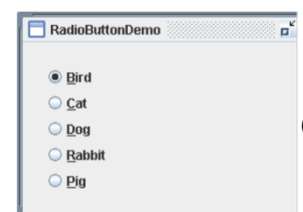
## Merkelapper (labels) og tekstfelt – informasjonfelt

Merkelapper brukes for å vise statisk tekst i brukergrensesnittet, mens tekstfelt (disabled) brukes for å vise dynamisk tekst. For eksempel kan merkelapper brukes for å beskrive ulike kontrollelementer i brukergrensesnittet eller faste variabelnavn, mens disabled tekstfelt kan brukes for å beskrive resultatet til ulike målinger.



## Gjenkjennbarhet (F)

Konvensjoner sier også noe om hvordan ulike kontrollelementer skal se ut, fordi de må være gjenkjennbare. For eksempel skal radioknapper være små og runde, og ha en tydelig markert runding inni når de er valgt. Hvis man endrer utseende til kontrollelementer kan det gjøre brukeren usikker på hva kontrollelementet er og hvilke funksjon det har.



# F30 – Designprinsipper

Læringsmål:

- **Forstå hvilken funksjon designprinsipper har i design av grafiske brukergrensesnitt**
- **Få kjennskap til ulike sett med designprinsipper:**
  - **Don Normans seks prinsipper**
  - **Jacob Nielsens ti prinsipper**
  - **Prinsipper for universell utforming og WCAG**

## Flere typer designkunnskap (F)

Vi kan dele designkunnskap inn i:

- **Konvensjoner** = etablerte, anerkjente regler (normer) for hva komponenter brukes til i forbindelse med GUI-design. For eksempel brukes radioknapper når brukeren skal velge ett av flere alternativ.
- **Designprinsipper** = generelle «regler» (retningslinjer/heuristikker) som gjelder for «alle» produkter. De bygger på erfaringsbasert designkunnskap og er samlet inn over tid gjennom utvikling av ulike produkter/systemer
- **Teorier** = forklaringer på fenomener som ofte er basert på empiriske studier (dvs. observasjoner). For eksempel kan det inkludere konseptuelle og mentale modeller.

## Designprinsipper VIKTIG (F)

Designprinsipper er generelle retningslinjer som kan brukes for å oppnå godt design. De bygger på designkunnskap som er samlet over tid ved utvikling av ulike systemer. De har høy generaliserbarhet og kan anvendes på mange produkter og systemer.

Brukskvalitet er hvordan man oppnår anvendbarhet, effektivitet og tilfredshet for bestemte brukere med bestemte mål i bestemte brukskontekster. **Designprinsipper fungerer som sjekklister i designprosessen, og det kan redusere ressursbruken (eks: tid og penger) og antall designiterasjoner som trengs i utviklingen av nye designløsninger.** Bruk av designprinsipper kan føre til at man raskere kommer frem til en egnet designløsning, siden man innfører færre «brukbarhetsproblemer» i designet. **Det er likevel nødvendig å teste brukskvaliteten til en designløsning selv om man følger designprinsippene, fordi prinsippene er generelle retningslinjer for godt design og ikke en garanti.** Designprinsippene vil ikke fange opp spesielle aspekter ved brukskonteksten og de er ikke nødvendigvis lette å operasjonalisere. Designprinsippene vil altså ikke garantere god brukskvalitet, men de kan gjøre det lettere og raskere å oppnå det.

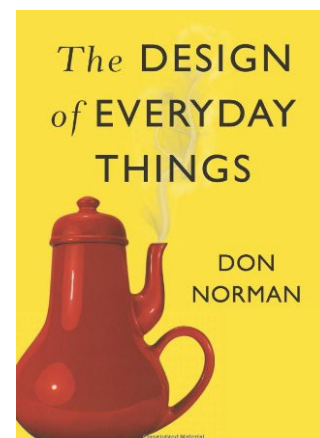


## Don Normans designprinsipper VIKTIG (F)

Don Normans prinsipper for *Design of Everyday Things* er:

- **Visibility** – kan jeg se det?
- **Affordance** – hvordan kan jeg bruke det?
- **Constraints** – hvorfor kan jeg ikke gjøre det?
- **Feedback** – hva gjør den nå?
- **Mapping** – hvor er jeg og hvor kan jeg gå?
- **Consistency** – jeg tror jeg har sett dette før?

Vi skal se nærmere på disse.





## Visibility – «Kan jeg se det?»

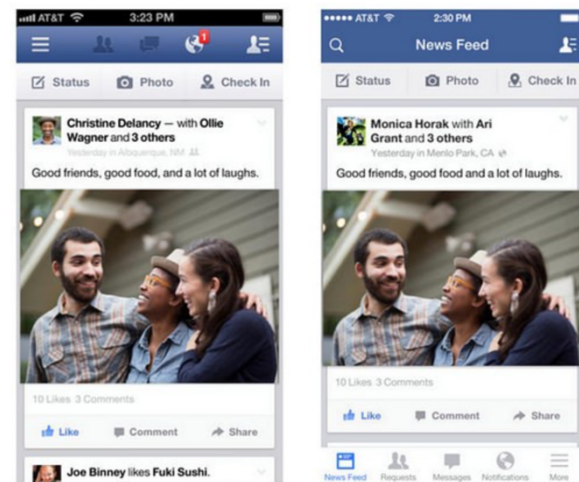
**Visibility** går ut på at jo mer synlig funksjonaliteten er, desto mer sannsynlig er det at brukerne forstår hva de kan gjøre. Høy synlighet gjør det lettere for brukere å oversette mål til aksjoner og det blir lettere for brukeren å se tilstanden. **Hvis en funksjonen er skjult vil det være vanskeligere å finne den og vite hvordan den skal brukes.** Et eksempel på dårlig visibility er en automatisk vannkran der synlige kontrolelementer er erstattet med (usynlige) aktive soner som kan få brukeren til å spørre seg selv: «Hvordan slår jeg på vannet?» eller «Hvor skal jeg putte hendene?». Dette kan være frustrerende for brukerne.

**I et GUI vil funksjoner som er ut av syne, for eksempel i menyer, faner og views, være vanskeligere å finne og dermed blir det vanskeligere å forstå hvordan de skal brukes.**

Aspekter som kan føre til dårlig visibility er:

- **Begrenset skjermstørrelse** – i apper på smarttelefoner kan det hende brukeren må sveipe mellom ulike skjermer, noe som gjør at ulike deler av elementene blir skjult
- **For mange elementer på skjermen** – brukeren har begrenset evne til å fokusere på elementer, så for mange elementer vil redusere synligheten

Et eksempel på visibility er hamburgermenyen på Facebook. Det er en nyttig plass å lagre ulike meny-enheter på en mobilapplikasjon, men den mangler visibility. Facebook har byttet fra hamburgermenyer til en fanemeny på bunnen av siden, for å øke synligheten av deres hovedenheter. Det er ikke alle elementer som kan bli synlige, fordi dette vil føre til for mange elementer på grensesnittet. Derfor er det **nødvendig med en prioritering, der størst visibility gis til elementene som er viktigst for brukeropplevelsen.**



## Affordance – «Hvordan kan jeg bruke det?»

**Affordance er hvilken handling en gjenstand signaliserer gjennom sin fysiske utforming. Ting vil «invitere» til visse handlinger eller bruksmuligheter,** for eksempel vil et dørhåndtak si «trykk med ned», en legokloss sier «sett oss sammen» og en drikkeflaske sier «drikk av meg». Brukere vil gjenkjenne og tolke signalene umiddelbart. Affordance beskriver relasjonen mellom utformingen til et objekt og de fysiske egenskapene til aktøren. **Affordance er de mulige aksjonene objektet gir aktøren,** for eksempel vil et tre ha følgende affordance for en apekatt: «kan klatres i», «gir beskyttelse» og «gir en mulighet for å klø seg». **Affordance vil avhenge av aktøren,** for eksempel vil ikke treet ha samme affordance for en hest som for en apekatt. Begrepet ble raskt brukt i interaksjonsdesign, der **hensikten med affordance er å gjøre det åpenbart for brukeren hvordan et produkt kan brukes.**



Vi skiller mellom to typer affordance:

- **Real affordance** = hva aktøren faktisk kan gjøre med et fysisk objekt (eks: gripe). Det er ofte åpenbart hva vi kan gjøre med dem, så opplæring er ikke nødvendig
- **Perceived affordance (brukt av Norman)** = hva aktøren tror hen kan gjøre med et (virtuelt) objekt. For eksempel «er dette en knapp?», «kan jeg dra dette elementet?» eller «kan jeg scrolle eller sveipe?». Dette kan være utfordrende å kommunisere via en skjerm, siden den kun har to dimensjoner (kan ikke ha real affordance).

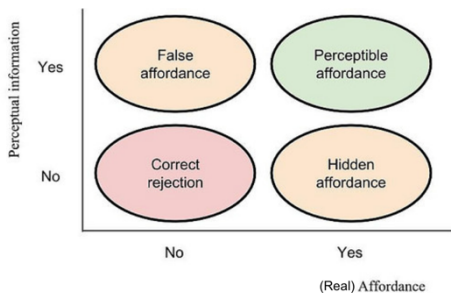
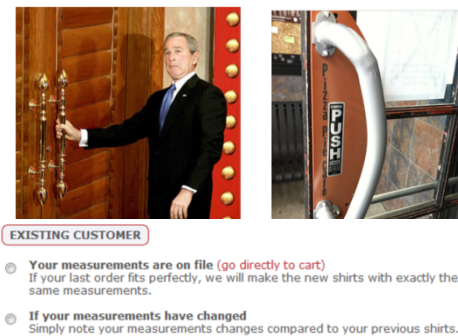
Merk: brukergrensesnitt som er skjermbaserte er virtuelle og kan ikke ha real affordance. Vi bruker perceived affordance i design av grensesnitt og dette er lærte konvensjoner.

**God affordance betyr at objektet gir et hint om hvordan det skal brukes.** Det vil være veldig klart hvordan man skal bruke et objekt med sterk affordance. Eksempler på god affordance er volum-sliden, tidspunkt-spinneren og unlock-sliden, siden de gir gode hint om hvordan de skal brukes. Submit-knappene viser god kontra dårlig affordance. Knappen til venstre har god affordance fordi den har en tydelig ramme og dimensjon, slik at det klart for brukeren at dette er en knapp som skal trykkes på. Knappen i midten og til høyre har dårlig affordance fordi de mangler trekkene som gjør det klart at det er en knapp, og dermed blir det vanskeligere for brukeren å finne ut hvordan de skal bruke elementet.



Motpartene til god affordance er:

- **Falsk affordance** = aktøren oppfatter at en handling kan utføres, når den i virkeligheten ikke kan det. Noen årsaker:
  - **Ingen sammenheng mellom handlingen objektet inviterer til og faktisk funksjon.** Eksempler er dører som kun er kulisser (dvs. kan ikke åpnes) og håndtak som signaliserer trekk på en dør som må skyves.
  - **Objektet signaliserer feil handling.** For eksempel merkelapper som ser ut som knapper eller et aktivt alternativ i menyen (eks: «Velg meg») som ikke reagerer.
- **Hidden affordance** = en handling som kan utføres i virkeligheten oppfattes ikke av aktøren. Årsak er:
  - **Objektet signaliserer ikke handlingen.** Eksempler er skjulte dører og knapper som ser ut som merkelapper



Denne figuren illustrerer de ulike tilfellene.

1. **God (perceptible) affordance** = handling oppfattes av bruker og handling kan utføres. Eks: knapp som responderer på klikk.
2. **False affordance** = handling oppfattes av bruker, men handling kan ikke utføres. Eks: knapp som ikke responderer på klikk.
3. **Hidden affordance** = handling oppfattes ikke av bruker, men handling kan utføres. Eks: ikke-knapp som responderer på klikk
4. **Riktig avslag** = handling oppfattes ikke av brukeren og handling kan ikke utføres. Eks: ikke-knapp som ikke responderer på klikk

### Affordance og signifiers

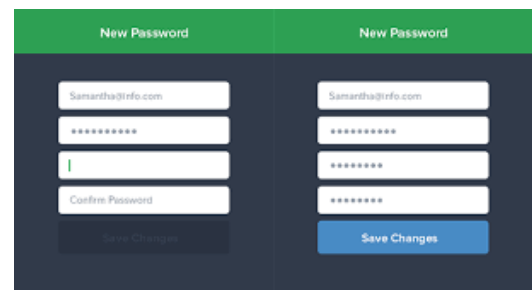
Norman skilte mellom:

- **Affordance** = brukernes oppfattelse av fysiske egenskaper til objekter og hva de kan gjøre med dem.
- **Signifiers** = andre, gjerne sosiale indikatorer, som styrer brukernes oppførsel og interaksjon. For eksempel vil en tom togperrong fortelle brukeren at toget har gått, et bokmerke forteller leseren hvor mye det er igjen av historien og antall «likes» kan indikere hvor mye et innlegg har blitt delt.

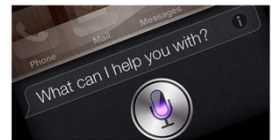
## Constraints – «Hvorfor kan jeg ikke gjøre det?»

**Constraints signaliserer begrensninger i forhold til handlinger, altså hva som ikke kan gjøres med et produkt.** For eksempel

kan lego-klosser bare settes sammen på bestemte måter, kantsteiner langs veien bestemmer hvor vi kjører, knapper som er «grået ut» kan ikke trykkes på og datofelt kan evt. bare tillate tall og punktum. **Constraints handler om å begrense interaksjonsmulighetene til brukeren for å forenkle grensesnittet og veilede brukeren til neste passende handling.** Det er viktig at begrensningene er klare og at brukerne får tydelig informasjon om hva som kan gjøres, slik at de ikke blir forvirret.



Et eksempel på manglende constraints er Siri-talesystemet på iPhone som ikke gir brukeren informasjon om hvilke spørsmål grensesnittet ikke støtter. Det kan være frustrerende for brukeren siden det blir vanskelig å vite hvordan systemet kan brukes.



## Affordance og Constraints

**Affordance og Constraints utfyller hverandre og kan sammen føre til godt design, siden riktig constraints begrenser handlinger som ikke kan utføres og god affordance gjør det lett å oppfatte handlinger som kan utføres.** Et eksempel på dette er dører der en metallplate brukes på *push*-siden og et dørhåndtak brukes på *pull*-siden. Et annet eksempel er søppelbøtter der formen til åpningen begrenser hvilke type søppel

som kastes hvor og selve hullet signaliserer at søppel kan kastes der. Et annet eksempel er oppsett av mail, der kanten og dimensjonen til create knappen gjør at brukeren oppfatter at denne skal trykkes inn, men knappen blir først tilgjengelig når feltene er utfylt.

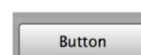


Affordance vil foreslå de mulige handlingene, mens constraints vil begrense antall alternativer. **For å sikre god affordance kan det hjelpe å bruke ulike typer constraints:**

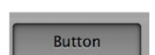
- **Fysiske begrensninger** – kroppens anatomi og gjenstandens form vil kunne begrense hva som kan gjøres med objektet. For eksempel kan nøkler kun settes inn vertikalt.
- **Semantiske og kulturelle begrensninger** – lærte konvensjoner, for eksempel har vi lært at rødt lys mener stopp og grønt lys mener kjør, slik at vi kommer frem til at rødt betyr at enheten er passiv og grønt betyr at den er aktiv. De fleste vet at man står forovervendt i heisen, ikke snakker i mobilen under forelesning, osv.
- Logiske begrensninger = innebærer resonering, for eksempel bør ordningen av knappene på komfyren matche plasseringen av komfyrplatene og knappene som er plassert på en dør bør brukes for ulike funksjoner på dørene.

## Feedback – «hva gjør den nå?»

**Feedback brukes for å gi brukeren informasjon om hvilke aksjoner som har blitt utført og hvilken tilstand systemet er i.** Feedback vil «fullføre» affordance, ved at affordance gir informasjon om handlinger som kan gjøres med et objekt og feedback gir informasjon om at handlingen er utført. For eksempel hos en knapp vil definert kant og dybde gjøre at det ser ut som en knapp (= affordance), mens når den endrer seg ved klikk vil brukeren se at den oppfører seg som en knapp (= feedback).



→  
(Klikk)



«Det ser ut som en knapp»  
(*affordance*)

«Det oppfører seg en knapp»  
(*feedback*)

Feedback handler om å gi brukeren informasjon om hvilke handlinger som har blitt utført og hva som har blitt oppnådd. **Gyldige handlinger bør alltid gi feedback**, for eksempel når du trykker på knappen skal rammen endres, når du knepper en jakke sier det «knepp» eller når blinklyset slås på vises det på dashbordet. **Mangel på feedback vil gjøre brukeren usikker**, for eksempel prosesser uten tilstandsindikator (progressbar) eller manglende kvittering etter brukeren har trykt på «betal»- eller «send-inn»-knapp.



Ulike typer feedback:

- **Visuell** – musepeker, progressbar, knapp som trykkes ned, osv.
- **Auditiv** – knappe-knepp, feil-beep, varsel ved mottatt SMS eller epost, osv.
- **Taktil** (berøring) – varierende motstand, vibrasjon, osv.

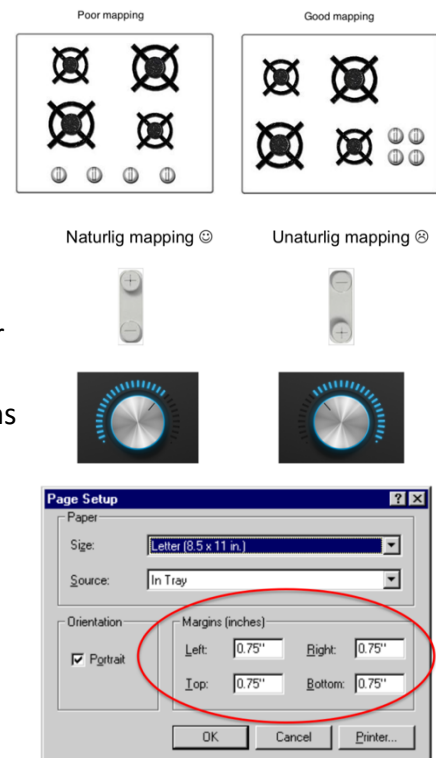
Tilbakemeldingen kan også være en kombinasjon av disse.

### Mapping – «hvor er jeg og hvor kan jeg gå?»

**Mapping handler om sammenhengen mellom kontrollelementer og deres effekt, spesielt mht. plassering og interaksjonsmuligheter.**

Målet er å ha en naturlig mapping, for eksempel kan vi se på forholdet mellom kontrollere og stekeflatene hos en stekeplate. Andre eksempler på mapping er:

- Volumkontroll med knapper over hverandre: øverste knapp skrur volum opp, mens nederste knapp skrur volum ned
- Volumkontroll med vri-knapp: med klokka gir høyere volum, mens mot klokka gir lavere
- Bilratt: mot venstre svinger bilden til venstre, mens mot høyre svinger bildet til høyre
- GUI: venstre/høyre- og topp/bunn-verdier bør plasseres sammen i hver sin vertikale liste.



Konfigurering av side-oppsett

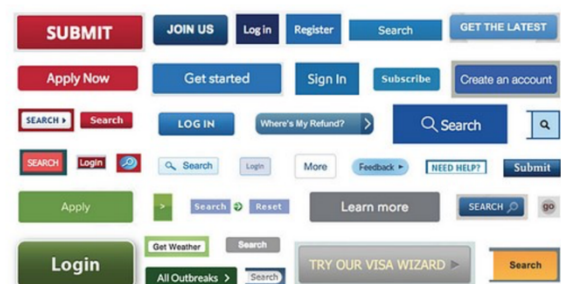
### Consistency – «jeg tror jeg har sett dette før?»

**Consistency handler om at elementer som ser like ut bør ha samme funksjon, altså at oppførselen til en bestemt oppgave bør være konsistent.**

For eksempel kan vi velge objekter i et grensesnitt ved å trykke på venstre museknapp (ikke høyer). **Consistency lar brukeren benytte seg av tidligere erfaring med lignende produkter, slik at det blir enklere å lære hvordan produktet brukes.**

**Inkonsistens skaper forvirring og bruker må huske unntak**, noe som vil redusere brukskvaliteten og dermed også brukeropplevelsen.

**Konvensjonene til kontrollelementene må følges** og konsistens er viktig på tvers av plattformer (OS X, Windows, IOS, Android, osv.). Figuren viser eksempler på dårlig consistency.



### 1.6.3 Don Normans designprinsipper (B)

Designprinsipper brukes av interaksjonsdesignere for å gjøre det lettere å designe for gode brukeropplevelser. Det er generaliserte retningslinjer som skal få designeren til å tenke på ulike aspekter av deres design. Designprinsipper stammer fra en blanding av teori, erfaring og sunn fornuft, og de vil ofte gi DOs og DON'Ts ved interaksjonsdesign. De vil ikke gi

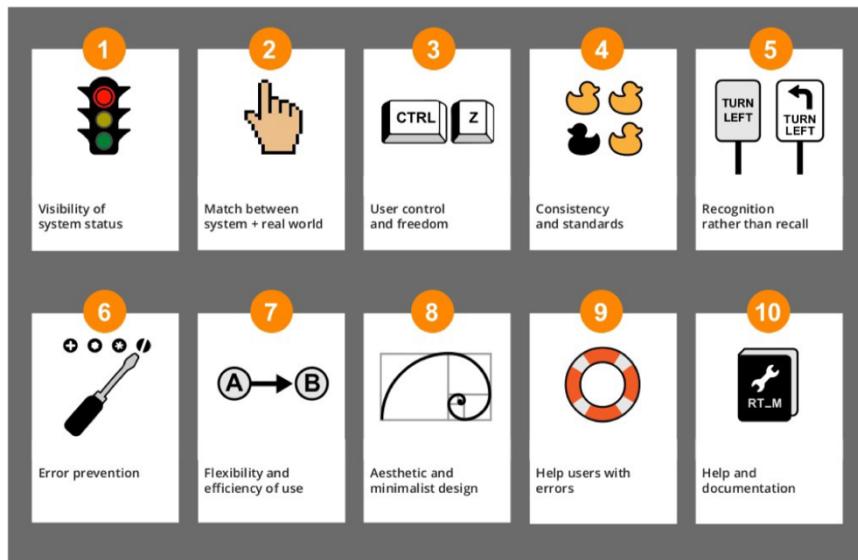
hvordan grensesnittet skal designes, men heller hjelpe designere til å forstå hva de kan forbedre med deres design. Don Normans designprinsipper består av følgende:

- **Visibility – jo mer synlig funksjonene er, desto mer sannsynlig er det at brukerne vil vite hva de skal gjøre videre.** I eksempelet med telefonen på hotellet vil mottatte anrop være skjult, mens ved klinkekule svarmaskinen vil de være synlige som klinkekuler. Kontrollene i en bil (eks: horn, varselampe, osv.) er synlige og indikerer derfor hva som kan gjøres.
- **Affordance – egenskaper ved et objekt gjør at brukerne får en oppfatning av hvordan objektet skal brukes.** Objektet gir et hint om hvordan det skal brukes. For eksempel vil et dørhåndtak gi hint om at den kan dras. Ved interaksjonsdesign vil affordance bety at det skal være åpenbart hva som kan gjøres med et grensesnitt. For eksempel må GUI-kontrollelementer som knapper, ikoner, lenker og scrollbarer formes slik at det er åpenbart hvordan de bør brukes.
- **Constraints – begrenser hvilke typer brukerinteraksjoner som er tillatt ved et gitt øyeblikk.** Dette kan oppnås på flere ulike måter, for eksempel kan kontrollelementer deaktiveres ved å farge dem grå, slik at brukeren ikke får benytte seg av disse ved dette steget i aktiviteten. Dermed hindrer man brukeren fra å velge feil alternativ og sannsynligheten for å gjøre feil blir redusert. Grafiske representasjoner kan også begrense brukerens tolkning av et problem, for eksempel kan flytdiagram vise hvilke objekter som er relatert, slik at tolkningen av informasjonen blir begrenset. Det fysiske designet til enheten kan også begrense hvordan det brukes (eks: USB-inngang på PC)
- **Feedback – brukerne må få informasjon om hvilken handling som har blitt utført og hva som har blitt oppnådd, slik at de kan fortsette med aktiviteten.** Flere typer feedback er tilgjengelig, for eksempel audio, taktil, verbal, visuell og kombinasjoner av disse. Riktig bruk av feedback kan gi nødvendig visibility for brukerinteraksjonen.
- **Mapping - sammenhengen mellom kontrollelementer og deres effekt, spesielt mht. plassering og interaksjonsmuligheter.** Forholdet mellom plasseringen av kontrollene i bilen og deres effekt gjør at det blir lett for sjåføren å finne riktige kontrollere
- **Consistency – i et grensesnitt bør like operasjoner og like elementer brukes for å oppnå like oppgaver. Et konsistent grensesnitt følger regler og konvensjoner,** for eksempel vil venstre klikk brukes for å velge et objekt på grensesnittet. Inkonsistens kan være vilkårlig, noe som gjør at det blir vanskelig for brukeren å huske handlingen og dermed øker sannsynligheten for feil. Fordelen ved konsistente grensesnitt er at de er enklere å lære og bruke, siden det brukeren lærer for en handling vil gjelde for alle lignende handlinger. Det kan være vanskelig å oppnå for kompliserte grensesnitt.

Et problem ved å bruke designprinsipper er at det kan oppstå byttehandler, for eksempel kan økt constraints på et grensesnitt føre til mindre visibility. Dette kan også skje ved bruk av ett designprinsipp, for eksempel kan et forsøk på å øke affordance ved å få objektene til å ligne fysiske objekter føre til at grensesnittet blir rotete og vanskelig å bruke. Hvis grensesnittet er konsistent med noe, kan det gjøre at det er inkonsistent med noe annet. Consistency kan også føre til at grensesnittet blir vanskeligere å bruke. For eksempel vil smørkniver plasseres i skuffen, brødkniver og skarpe kniver i en holder, jack-kniv i sekken, osv. Plasseringen er inkonsistent og det er mer krevende å lære, men knivene er plassert slik at de er lettere å finne i konteksten der de brukes. Inkonsistens kan gjøre det vanskeligere å lære grensesnittet, men lettere å bruke i lengden.

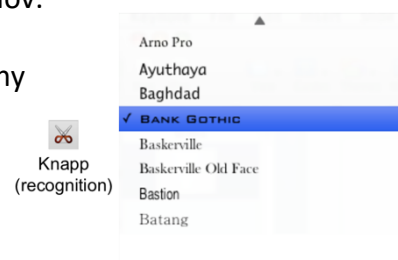
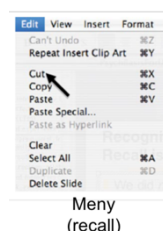
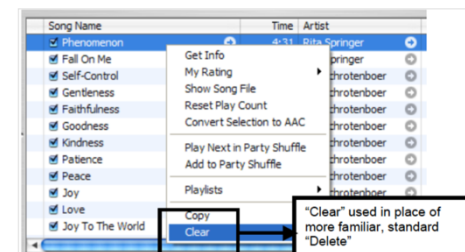
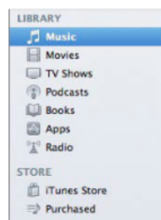
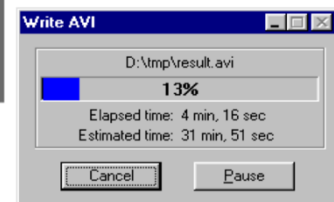
## Jakob Nielsens 10 brukbarhetsheuristikker VIKTIG (F, B)

Noen ganger er det utfordrende å utføre brukertester, for eksempel fordi brukere ikke er lett tilgjengelig. Derfor har ulike inspeksjonsmetoder blitt utviklet som alternativ til brukertester (eller til å brukes ved siden av). En av disse er heuristisk evaluering, der komponentelementer ved grensesnittet blir analysert og potensielle brukskvalitetsproblemer blir identifisert vha et sett med retningslinjer. Det kalles heuristikk siden retningslinjene er generelle tommelfingerregler og ikke spesifikke måter å oppnå høy brukskvalitet. Disse retningslinjene ligner designprinsipper:

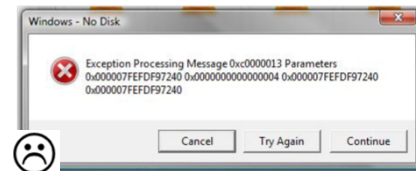
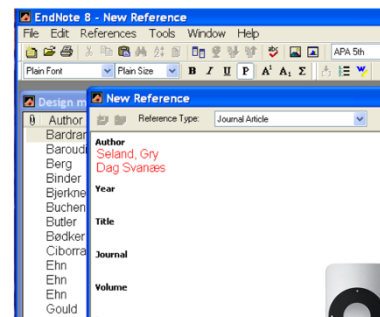


Jakob Nielsens 10 brukbarhetsheuristikker er:

1. **Visibility of system status** – systemet skal alltid informere brukeren om hva som foregår i maskinen vha. passende feedback innen rimelig tid.
2. **Match between system and the real world** – systemet skal snakke **brukerens språk**, med begrep, ord og fremstillingsmåte (metaforer) som brukeren forstår. Følg konvensjoner fra den virkelige verden, slik at informasjonen får en naturlig og logisk orden. For eksempel blir butikk (iTunes store) brukt for å kjøpe nytt medieinnhold som kan legges til biblioteket.
3. **User control and freedom** – brukere vil ofte gjøre feil og trenger tydelige markerte måter å forlate en uønsket tilstand så raskt som mulig. Systemet må altså støtte undo og redo, og brukeren skal ha kontroll. Alle nivåer må implementere angre-mulighet.
4. **Consistency and standards** – brukere skal slippe å vurdere om ulike ord, situasjoner eller aksjoner betyr det samme. Konvensjonene (retningslinjene) til plattformen må følges, slik at grensesnittet blir konsistent. For eksempel må man være konsistent i bruk av Delete/Clear og New/Create.
5. **Recognition rather than recall** – gjør objekter, handlinger og alternativer synlige, slik at brukeren ikke trenger å huske all informasjon. Instruksjoner bør være synlige eller lette å hente ved behov. Kunnskap skal være i systemet og ikke i hodet. Eksempler er forhåndsvisning ved valg av font eller bruk av knapp istedenfor meny eller hurtigtast.



6. **Error prevention** – lag systemet slik at minst mulig feil er mulig. Fjern deler som er utsatt for feil eller tillatt fleksibilitet, for eksempel at både fornavn og etternavn kan skrives inn først.
7. **Flexibility and efficiency of use** – tilby avanserte snarveier og hurtigtaster for handlinger som utføres ofte. For eksempel vil cmd+C være kopiering i OS X. Disse er usynlige for nye brukere, men vil gjøre interaksjonen raskere for ekspertbrukere, slik at systemet kan brukes av både erfarne og uerfarne brukere.
8. **Aesthetic and minimalist design** – dialoger bør ikke inneholde irrelevant informasjon. For mye informasjon reduserer synligheten, så alt som ikke tjener en hensikt skal vekk! Systemet skal være estetisk fornøylig og effektivt satt sammen
9. **Feilhåndtering** – hjelp brukere til å unngå og håndtere feil ved å gi feilmeldinger med normalt språk (ingen koder) som presist indikerer problemet og foreslår en løsning
10. **Help and documentation** – lag et godt hjelpesystem med hurtigreferanser, brukermanual og verktøytips. Selv om det er bedre om systemet kan gis uten dokumentasjon, kan det være nødvendig for å gi brukeren hjelp. Denne informasjonen skal være lett å søke gjennom, fokusert på brukerens oppgaver, liste konkrete steg som kan utføres og ikke være for stor.



### 15.1.15.2 Heuristisk evaluering (B)

Ved evaluering vil en eller flere eksperter analysere grensesnittet ved at grensesnittelementene blir inspisert og sammenlignet med listen av heuristikker. Hver analyse kan identifisere brukskvalitetsproblemer og prosessen gjentas til de fleste er ordnet. Jo flere eksperter desto mer sannsynlig er det at problemer avdekkes. Det er en rask evaluering (krever ikke brukere eller spesielle fasiliteter) som ofte brukes tidlig i designfasen.

### Universell utforming – 7 prinsipper (F)

**Universell utforming er design og sammensetning av et miljø som kan nås, forstås og brukes av alle mennesker uansett alder, størrelse, evner eller funksjonshemninger.** Universell utforming bygger på **Design for all** og et **inkluderingsprinsipp** der omgivelser, produkter og tjenester blir muliggjort for alle mennesker. Det er ikke et spesielt krav som kun lønner en del av populasjonen, men et grunnleggende krav for godt design. Dette prinsippet har et **etisk grunnlag** og det danner basisen for utarbeidelsen av WCAG-anbefalingene som sier noe om tilgjengeligheten for webinnhold.



**Universell utforming tar utgangspunkt i et menneskelig mangfold** som består av mennesker med ulike fysiske, kognitive og sensoriske evner og inkluderer barn, voksne, eldre, friske, syke, menn, kvinner og ulike kropper. **Ved å designe for et menneskelig mangfold kan vi lage produkter som er enklere for å alle å bruke.** Funksjonshemming oppstår når det ikke er samsvar mellom funksjonsevnen til en person og kravene som stilles av omgivelsene. Funksjonshemmingen er ikke en del av personen, men er heller relasjonell og skapes av omgivelsene. Det kan for eksempel være et resultat av dårlig brukskvalitet, så funksjonshemming kan ses på som et designproblem.

De 7 prinsippene for universell utforming er:

1. **Like muligheter for bruk – designet skal kunne brukes av mennesker med ulike evner.** Ulike brukergrupper skal få samme muligheter til bruk. Løsningene skal være like hvis det er mulig eller likeverdige hvis ikke. Det er viktig å unngå segregering og stigmatisering av brukere, slik at privatliv, sikkerhet og trygghet blir like tilgjengelig for alle.



Automatiske skyvedører

2. **Fleksibel i bruk – designet skal tjene et vidt spekter av individuelle preferanser og evner.** Designet lar brukere velge metode, det skal tjene både høyre- og venstrehandsbruk, gjøre det lettere for brukeren å være nøyaktig og presis og tilpasse seg brukerens hastighet.



Saks for venstre- og høyre hentede brukere.

3. **Enkel og intuitiv bruk – designet skal være enkelt å forstå uten hensyn til brukerens erfaring, kunnskap, språkferdigheter eller konsentrasjonsnivå.** Dette innebærer at designet eliminerer unødvendig kompleksitet, stemmer overens med brukerens forventninger, tjener ulike lese-, skrive og språkferdigheter, ordner informasjon basert på hvor viktig det er og gir tilbakemeldinger under og etter fullføring av en oppgave.



Brukermanualer med bilder i stedet for tekst.



Pasientalarm med trykk-knapp

4. **Forståelig informasjon – designet skal kommunisere nødvendig informasjon til brukeren på en effektiv måte, uavhengig av forhold knyttet til omgivelsene eller brukerens sensoriske evner.** Dette innebærer å bruke ulike måter å representere essensiell informasjon (bilder, verbalt, taktilt) og det må være en kontrast mellom informasjonen og omgivelsene. Lesbarheten av essensiell informasjon må maksimeres og det legges til rette for teknikker og innretninger som brukes av mennesker med sensoriske begrensninger. For eksempel vil flytterminaler ha en bred presentasjon av info (tavler, høytaleranlegg, personell, osv.)



Flytterminaler har bred presentasjon av essensiell informasjon: Informasjonstavler (visuelt), høytaleranlegg (auditiv, flere språk) og personell

5. **Toleranse for feil – designet skal minimalisere farer og skader som kan gi ugunstige konsekvenser.** Dette krever at elementene ordnes slik at de mest brukte blir mest tilgjengelig, mens de farlige blir eliminert, isolert eller skjult. Designet må gi advarsler om farer og feil, tilby sikkerhetstjenester og oppmuntre bevisst handlinger.



6. **Lav fysisk anstrengelse – designet kan brukes effektivt og komfortabelt med minimum anstrengelse.** Dette innebærer at brukere skal kunne ha en nøytral kroppstilling og bruke rimelig betjeningsstyrke. Designet må minimere gjentakende handling og kreve liten vedvarende fysisk kraft.



Berøringssensitive lamper erstatter på/av-bryter



Trådløse betalingsterminaler.

7. **Størrelse og plass for tilgang og bruk – hensiktsmessig størrelse og plass skal muliggjøre tilgang, rekkevidde, betjening og bruk, uavhengig av brukerens kroppsstørrelse, kroppstilling eller mobilitet.** Dette innebærer å



- Klar synslinje for sittende og stående  
- Fysisk plass som muliggjør tilgang og bruk



muliggjøre en klar synslinje og bekvem rekkevidde til viktige elementer for både sittende og stående brukere. Designet må støtte variasjoner i hånd- og gripsstørrelse og det må legges til rette for nok rom for bruk av hjelpemidler og personlig assistanse.

## WCAG-retningslinjer

**WCAG definerer hvordan webinnhold kan gjøres mer tilgjengelig for mennesker med nedsatt funksjonsevne** (eks: blinde, døve, personer med lærevansker, kognitiv funksjonsnedsettelse, talevansker, osv.). Når retningslinjene følges, kan webinnholdet bli mer brukervennlig. Det brukes blant annet for å gjøre webinnholdet mer brukervennlig for eldre personer med funksjonsnedsettelse som er aldersbetingede.

På øverste nivå består WCAG av 4 prinsipper som danner grunnlaget for tilgjengelighet på nett:

- 1. Mulig å oppfatte** – informasjon og grensesnittkomponenter må presenteres for brukere på måter som de kan oppfatte. Informasjonen som presenteres kan ikke være usynlig for alle deres sanser.
- 2. Mulig å betjene (bruke)** – grensesnittet kan ikke kreve interaksjon som en bruker ikke kan utføre
- 3. Forståelig** – innholdet eller interaksjonen kan ikke være utenfor forståelsen til brukerne
- 4. Robust** – innholdet må kunne tolkes av et bredt spekter med brukere og brukere må få tilgang til innhold etterhvert som teknologier utvikler seg.



Hvis noen av disse ikke er sanne, vil ikke brukere med funksjonsnedsettelse kunne benytte seg av webinnholdet.

**På nivået under prinsippene er det 12 retningslinjer.** Disse utgjør de grunnleggende målene som produsenter av webinnhold bør jobbe mot for å gjøre innholdet mer tilgjengelig for brukere med ulike funksjonsnedsettelse. Selve retningslinjene kan ikke testes, men hver retningslinje har en eller flere suksesskriterier som kan testes, slik at WCAG kan brukes for å sjekke at krav overholdes. For eksempel kan et suksesskriterie være «Med unntak av teksting og bilder av tekst kan tekst forstørres opptil 200 % uten at innhold eller funksjonalitet går tapt. (Nivå AA)». Suksesskriteriene deles inn i tre nivåer: A (lavest nivå), AA og AAA (høyest nivå). For eksempel kan det sies at nettsiden tilfredsstillende alle level AA suksesskriterier. Jo høyere nivå desto mer tilgjengelig er webinnholdet. De 12 retningslinjene er:

- 1. Mulig å oppfatte**
  - 1.1 Gi tekstalternativer til alt ikke-tekstlig innhold**, slik at det kan konverteres til formater som brukerne har behov for, for eksempel stor skrift, blindeskrift, tale, symboler eller enklere språk
  - 1.2 Gi alternativer for tidsbaserte medier** (dvs. medier med kun lyd og/eller video, alternativer kan være teksting og tegnspråk)
  - 1.3 Lag innhold som kan presenteres på ulike måter uten at informasjon eller struktur går tapt** (eks: enklere layout)
  - 1.4 Gjør det enklere for brukere å se og høre innhold**, blant annet ved å skille forgrunnen fra bakgrunnen (dvs. kontrast)
- 2. Mulig å betjene**
  - 2.1 Gjør all funksjonalitet tilgjengelig med tastatur**
  - 2.2 Gi brukerne nok tid til å lese og bruke innhold** (eks: pause)

- 2.3 Ikke utform innhold på en måte som er kjent til å forårsake epileptiske anfall
- 2.4 Gjør det mulig for brukerne å navigere, finne innhold og vite hvor de befinner seg
- 3. Forståelig
  - 3.1 Gjør innholdet leselig og forståelig
  - 3.2 Sørg for at nettsider presenteres og fungerer på forutsigbare måter
  - 3.3 Hjelp brukere med å unngå feil og å rette opp feil
- 4. Robust
  - 4.1 Sørg for best mulig kompatibilitet med aktuelle og fremtidige brukere, inkludert kompenserende teknologi

For hver retningslinje og hvert suksesskriterium er det dokumentert en rekke teknikker. Disse er informative og kan deles inn i to kategorier: teknikker som er tilstrekkelige til å overholde suksesskriteriene og teknikker som er veiledende. De veiledende teknikkene er mer omfattende enn det som kreves av de enkelte suksesskriteriene, og gir dermed produsenter av webinnhold mulighet til å gjøre bedre bruk av retningslinjene. **Alle disse veiledningsnivåene (dvs. prinsipper, retningslinjer, suksesskriterier og teknikker) gir råd om hvordan innhold kan gjøres mer tilgjengelig.** Produsenter av webinnhold oppfordres til å bruke så mange nivåer som mulig for å dekke behovene til flest mulig brukere.

# F40 – Gestaltprinsipper

Læringsmål:

- Forstå hva gestalt-teorien sier om hvordan mennesker tolker visuelle inntrykk
- Få kjennskap til fem gestalt-prinsipper: likhet, nærhet, mental komplettering, kontinuitet og forgrunn/bakgrunn
- Forstå relevansen av gestaltprinsipper for design av grafiske brukergrensesnitt

## Gestaltpsykologi <sup>(F)</sup>

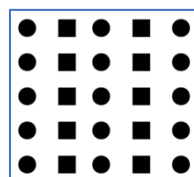
**Gestalt er tysk for form eller figur, og i følge gestaltteorien tolker mennesker sansenintrykk som meningsfylte helheter. Brukerne vil oppfatte sammenhenger fremfor enkeltdele.** Persepsjon er sanseoppfatninger og den påfølgende tolkningen, og det innebærer to trinn: stimulering av et sanseorgan og resulterende opplevelse. For eksempel kan vi oppfatte en vedvarende, dyp lyd utenfor og tolke dette som et jagerfly. Gestaltpsykologi er et studie av persepsjon som ser på hvordan vi sanser og organiserer visuelle inntrykk som helheter. **Det perseptuelle systemet organiserer sansenintrykk i helheter**, for eksempel vil en melodi oppfattes som en helhet i stedet for en serie av enkelttoner, mens på bildet vil vi se en hund og ikke enkelte svarte og hvite felter. Gestaltprinsippet kan summeres som: «**The whole is *other* than the sum of the parts**».



## Gestaltprinsipper <sup>(F)</sup>

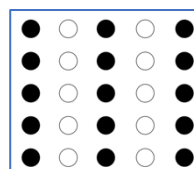
**Gestaltprinsipper er et sett med lover som beskriver hvordan mennesker kan oppfatte visuelle elementer ved å gruppere lignende objekter, gjenkjenne mønster og forenkle komplekse bilder.** Mennesker har et behov for å søke orden blant uorden, ved at minnet tolker en serie med elementer som et bilde eller en illusjon. Dette kan brukes i design for å engasjere brukerne. Designere bruker gestaltprinsipper for å lage design med vel plasserte elementer som fanger oppmerksomheten til brukerne. Det finnes mange gestaltprinsipper, men vi ser på:

- **Likhet** – like elementer kobles sammen
- **Nærhet** – nære elementer kobles sammen
- **Mental komplettering** – elementer oppfattes som hele
- **Kontinuitet** – mønster skaper forventning om fortsettelse
- **Forgrunn/bakgrunn** – objekter oppfattes å være i forgrunn eller bakgrunn
- **Symmetri** (ikke pensum) – designet bør være balansert og fullstendig



## Likhet

**Menneskeøyet søker etter ulikheter og likheter i et bilde og vil koble sammen like elementer.** Like former, farger og størrelser ses sammen. På figuren er det enklere å oppfatte de fem kolonnene sammenlignet med de fem radene.



Når man plasserer like objekter ved hverandre vil hjernen bruke tid og energi på å koble dem sammen, slik at den kan forstå deres forhold med hverandre. **I interaksjonsdesign kan likhet derfor brukes for å bygge koblinger mellom elementer som man ønsker å fremheve:**

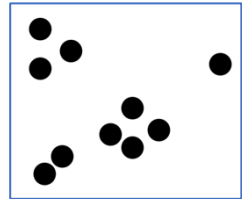
- **Innhold** = farge, fontstørrelse og type, osv. kan la brukeren skille mellom ulike typer innhold før de leser dem. For eksempel kan sitater markeres med «*sitat*».



- **Lenker** = brukes for å navigere mellom ulike sider og må presenteres på en lik måte slik at de kan identifiseres av brukeren. Lenker bør ha samme farge og form, for eksempel <https://www.vg.no>
- **Overskrifter** = brukes for å organisere innholdet og må ha ulik font, størrelse, farge, osv. sammenlignet med resten av innholdet. De hjelper brukeren til å finne relevante punkt i innholdet og kontrollere den totale flyten.

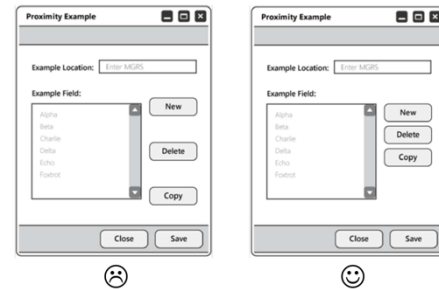
## Nærhet

**Elementer som står nære hverandre vil grupperes sammen, mens de som er lengre fra hverandre blir separert.** Enkle former grupperes for å skape mer komplekse bilder. På figuren vil vi se fire grupper med rundinger istedenfor 10 spredte. Øyet vil lage koblinger mellom like elementer, men dette vil også påvirkes av måten de er plassert i forhold til hverandre.



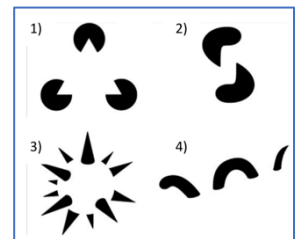
**I interaksjonsdesign kan nærhet brukes for å koble sammen eller separere elementer:**

- **Tekst** = mellomrom mellom tekstblokker sier noe om hvordan blokkene er relatert til hverandre. Avsnitt brukes for å samle tekst med lignende ideer, poenger og argumenter. Hvis du endrer tema må du starte et nytt avsnitt
- **Struktur** = elementer i grensesnittet kan grupperes ved å plasseres nære hverandre og gruppene skilles fra hverandre med mellomrom. Det er viktig å ikke plassere for mange elementer i en gruppe, for da vil strukturen bli uoversiktlig.



## Mental komplettering (lukning)

**Menneskeøyet foretrekker fullstendige former og vil derfor automatisk fylle inn gapene mellom elementer for å oppfatte fullstendige bilder.** Vi oppfatter helst lukkede eller sluttede former, og ser derfor det hele først (se figurer).



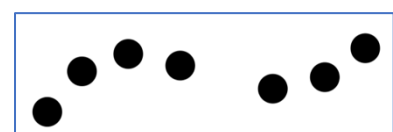
Når et bilde mangler informasjon vil øyet ignorere den manglende informasjonen og fylle gapene med linjer, farger eller mønster for å fullføre bildet. **I interaksjonsdesign blir mental komplettering brukt for å oppnå visuelle effekter:**

- **Logo** = brukes for å representere en bedrift, gruppe, osv. Mental komplettering kan brukes for å fange brukerens oppmerksomhet



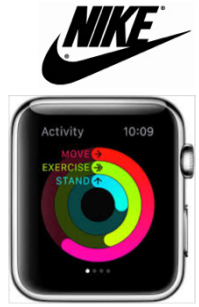
## Kontinuitet

**Menneskeøyet vil følge baner, linjer og kurver til et design og foretrekker å se en kontinuerlige flyt av visuelle elementer istedenfor separate objekter.** En linje som har en bestemt retning oppfattes som om den fortsetter i samme retning som den starter. Dette gjelder også mht. former. Den redundante informasjonen i mønsteret skaper forventingen om en fortsettelse.



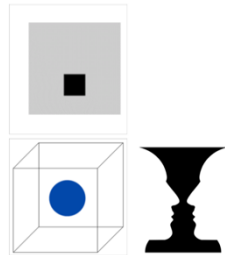
## I interaksjonsdesign blir kontinuitet brukt for å styre hvor brukerne ser:

- **Linjer** = brukeren ser at elementer på linjen er koblet sammen. For eksempel i en betalingsprosess kan linjer brukes for å koble sammen de ulike stegene og holde brukeren på en «bane» der de fortsetter til de når enden.
- **Logo** = kontinuitet kan brukes for å fange oppmerksomheten til brukeren.
- **Utseende til elementer** = mennesker foretrekker å se en kontinuerlig flyt av visuelle elementer istedenfor separate objekter

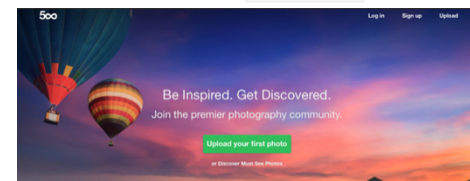


## Forgrunn/Bakgrunn

**Menneskeøyet vil instinktivt oppfatte at objekter enten er i forgrunnen eller bakgrunnen, basert på farge, kontrast, rammer, størrelse og perspektiv.** Ved et stabilt forhold mellom forgrunn og bakgrunn vil objektet lett separeres fra bakgrunnen og dermed dominere layouten. Ved reversibelt forhold kan øye flippe på figuren og bakgrunnen, slik at det lages en visuell illusjon, men det er alltid en klar forskjell mellom forgrunn og bakgrunn (se Rubin vase). Ved tvetydig forhold er denne forskjellen liten og et element kan være i begge samtidig.



- **Overskrift** = er som regel større enn resten av teksten, slik at overskriften oppfattes å være i forgrunnen, mens teksten er i bakgrunnen (stabilt forhold)
- **Tekst** = er som regel mørk, mens bakgrunnen er lys (eller motsatt), slik at teksten oppfattes å være i forgrunnen. Høy kontrast forbedrer lesbarheten (stabilt forhold) og omgivelsen vil påvirke opplevelsen av en farge.
- **Visuelle effekter** = forholdet mellom forgrunn og bakgrunn kan brukes for å fange oppmerksomheten til brukerne



## Bruk av gestaltprinsipper i GUI (F)

**Gestaltprinsippene forteller oss hvordan vi kan organisere elementer i et GUI slik at designet blir i tråd med menneskelig persepsjon. Det er viktig at brukerne forstår hva de ser og finner det de ønsker med ett blick.** For eksempel kan nærhet og likhet med farge og form brukes for å dele siden inn i separate deler. Uten dette vil brukeren slite med å tolke de ulike elementene og dermed bli frustrert. Gestaltprinsippene skal ikke forvirre eller forsinke brukere, men heller brukes for å veilede de. Prinsippene kan brukes for å lage mer interessante og engasjerende visuelle opplevelser for nettsider og applikasjoner.



# F41 – Konseptuelle- og mentale modeller

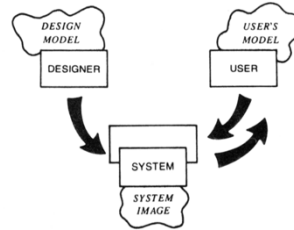
Læringsmål:

- Lære hva en konseptuell modell (designmodell) er i interaksjonsdesign
- Lære hva en brukers mentale modeller er i interaksjonsdesign:
  - Hva modellen påvirkes av og hvordan den endres
  - Hvordan vi som designere kan få aksess til en brukers mentale modell
- Forstå sammenhengen mellom de to modellene

## Tre modeller av et system

Vi skal se på tre modeller av et system:

- Konseptuell modell (designmodell)
- Mental modell
- System modell (systembilde)

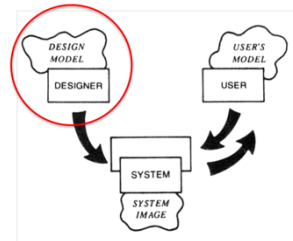


## Konseptuell modell (designmodell) <sup>(F)</sup>

**En konseptuell modell er en høy-nivå beskrivelse av hvordan systemet er**

**organisert og virker.** Det utgjør den mentale modellen som designerne ønsker at brukerne skal ha av virkemåten og strukturen til et system. Brukerne benytter den mentale modellen til å forutsi oppførselen til produktet og generalisere det de lærer til nye situasjoner. Den konseptuelle modellen beskriver hvordan

systemet er ment å se ut fra brukerens perspektiv. Produktet er rettet mot brukeren, så det er viktig å tenke på hvordan brukeren oppfatter produktet. Den konseptuelle modellen gir designerens ønske om hvordan produktet skal oppfattes av brukerne, strukturen til systemet (dvs. objekter og deres relasjoner) og mekanismene for hvordan systemet kan brukes for å oppnå oppgaver. **Modellen bør være så enkel som mulig (*less is more*) og oppgavefokusert (øker sannsynligheten for at bruker oppfatter produktet slik designeren ønsker).**



**Elementer i den konseptuelle modellen er:**

1. **Designmetafor** (hvis det benyttes) – i hvilken grad ligner brukergrensesnittet på noe brukeren har møtt før? Hva er likt og ulikt?
2. **Konsepter** – elementene brukerne må forholde seg til når de bruker systemet. Finn objekter og tilhørende attributter og operasjoner til disse. Forstår brukerne konseptene de eksponeres for? Snakker appen brukerens språk? Husk: *less is more*
3. **Relasjonene** – forhold mellom konseptene. Forstår brukerne relasjonene mellom konseptene de eksponeres for? Forstår de hva som fører til hva?
4. **Mapping** – overførbarheten mellom konseptene og brukskonteksten til systemet. I hvilken grad er konseptene overførbare til den virkelige verden? Hva korresponderer til hva?

## Eksempel 1 – online bibliotek katalog

For en online bibliotek katalog vil:

- **Designmetafor** = innholdet er ordnet som en fysisk katalog som kan søkes gjennom
- **Konsepter** = brukerkonto med navn og lånte elementer, bibliotekar, bok med tittel og forfatter, lånt element, osv.
- **Relasjoner** = bok lånt av bruker, osv.
- **Mapping** = hvert element i systemet korresponderer til et fysisk element i biblioteket

## Eksempel 2 – AtB Mobillett

For AtB Mobillett (bussbillett-app) vil:

- **Designmetafor** = innholdet er ordnet som en fysisk papirbillett der man kan velge antall og type, betale, motta og vise billetten (dette er likhetene)
- **Konsepter** = enkeltbillett der attributter er reiseform, type, gyldighet, pris, osv. og operasjoner er kjøpe, vise, endre sone, osv. Mobillett-konto der attributter er saldo, profil, bruker og operasjoner er fyll på saldo, legg til betalingskort, osv.
- **Relasjoner** = Mobillett-konto har en profil, har registrerte bankkort, har kjøpte billetter og registrerte brukere.
- **Mapping** = en Mobillett-konto korresponderer til én registrert bruker, et betalingskort registrert Mobillett-kontoen korresponderer til et fysisk bankkort, en gyldig billett registrert på en Mobillett-konto korresponderer til en gyldig tur på bussen.



## Hva en konseptuell modell ikke er

Den konseptuelle modellen til et interaktivt system vil ikke være:

- **Brukergransesnittet**, fordi modellen vil ikke gi layout i form av GUI-komponenter og deres plassering eller interaksjonsmetoder (touch, mus, tastatur, stemmeaktivering, osv.). Den vil kun beskrive hva brukerne kan gjøre med systemet og hvilke konsepter de må forstå for å operere det.
- **Brukerens mentale modell av systemet**, fordi den er ikke tilgjengelig for designerne og ulike brukere vil ha ulike mentale modeller.
- **Use cases eller scenarie-beskrivelser**, fordi use cases er historier om oppgaver som brukerne må utføre og skal uttrykkes uten hensyn til systemet. Use cases fokuserer på oppgaver, mens den konseptuelle modellen fokuserer på systemet
- **Objekt-orientert arkitektur**, fordi den konseptuelle modellen gir ingen tekniske detaljer om implementasjonen

## Valg av konseptuell modell

Den konseptuelle modellen påvirker brukerens forståelse av systemet, siden det bestemmer hvilke konsepter brukeren må forholde seg til, hva brukeren kan gjøre med konseptene og hvordan konseptene henger sammen. **Når man skal velge konseptuell modell vil det ofte innebære vanskelige avveininger, for eksempel enkelthet i bruk kontra rik funksjonalitet.**

## Konseptuell modell og designprosessen

Prosessen av å bruke en konseptuell modell er som følger:

1. **Lag den konseptuelle modellen** (ikke hvordan brukergrensesnittet skal se ut) ved å begynne med skjelettet/grunnmuren og gradvis forfine
2. **Lag brukergrensesnittet og test designet**, ved å for eksempel se om brukerne forstår konseptene de blir eksponert for og relasjonene mellom dem
3. **Gjør eventuelle endringer i konseptuell modell og i brukergrensesnitt, og test på nytt.** Dette gjøres **iterativt** til man er fornøyd med brukernes evalueringer. Det er kostbart å endre konseptuell modell i senere faser av designprosessen!

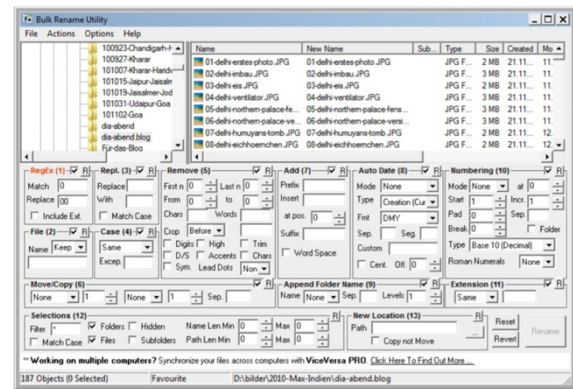
Kjennetegn ved en **god konseptuell modell** er:

- **Modellen «snakker» brukerens språk.** Hva elementer kalles i brukergrensesnittet er avgjørende for brukskvaliteten. Språk er viktig!

Task/Events	Without task-based model	With Task-based model
User wants to log a weeks worth of hours.	"Create record"	"Log hours"
User has successfully logged hours.	"Success: New row inserted"	"Hours successfully logged"
User forgets that he has already logged her hours	"Error: ORA-00001: unique constraint (CLEATS.PA_REPORT_HEADERS_U!) violated,"	"Hours for selected week has already been logged"



- Brukeren trenger ikke å forholde seg til mange konsepter samtidig
- Relasjoner mellom konseptene er åpenbare



På figuren kan vi se et eksempel på et brukergrensesnitt som er laget basert på en dårlig konseptuell modell. Det er veldig mange konsepter å forholde seg til, språket er teknisk (eks: Chars) og det er usikkert hvordan konseptene henger sammen.

### 2.1-2.3 Konseptuell modell (B)

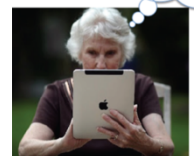
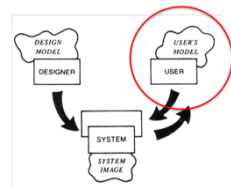
Dersom man har en klar forståelse for hvorfor og hvordan man kan designe noe kan man spare enorme mengder tid, innsats og penger senere i designprosessen. Lite gjennomtenkte ideer og ikke-kompatibel og ubrukelig design kan raffineres mens det fortsatt er lett. Det blir lettere å fokusere på brukskvalitet og brukeropplevelsen. Før man bestemmer hvilken teknologi som skal brukes og hvordan aspekter skal designes, bør man uttrykke egenskapene til problemrommet. Dette krever at designerne gjør antagelser og påstander som må forsvares, noe som kan føre til at man oppdager problemer ved designløsninger. Forståelsen av problemrommet kan så brukes for å lage en konseptuell modell.

En konseptuell modell er en abstrakt skisse for hva brukerne kan gjøre med produktet og hvilke konsepter som trengs for å forstå hvordan man interagerer med det. **De ulike elementene er metaforer, konsepter, relasjoner og mapping, og hvordan disse er organisert vil bestemme brukeropplevelsen.** Den konseptuelle modellen kan brukes av designteamet for å finne designløsningen som er enklest, mest effektiv og lettest å lære/huske. **Produktet vil ha høy brukskvalitet dersom den mentale modellen til brukeren i stor grad overlapper designerens konseptuelle modell.**

### Mental modell (F)

**En mental modell er den individuelle brukerens tankegang om hvordan systemet fungerer, og den er basert på ufullstendig fakta, tidligere erfaringer og intuitiv oppfatning.** Brukerens mentale modell er altså hva en bruker tror om virkemåten og strukturen til systemet, og den **vil i stor grad kontrollere hvordan brukeren interagerer med systemet.** Modellen former aksjonen. For eksempel hvis en eldre person bruker en iPad, kan deres mentale modell være basert på tidligere erfaring med bøker, slik at de interagerer med iPaden som om den var en bok. Tre egenskaper ved mentale modeller:

- **Subjektive** – modellen hører til den individuelle brukeren og ulike brukere vil ha ulike mentale modeller
- **Dynamiske** – modellen vil formes ettersom brukeren tolker visuell struktur og synlig oppførsel hos systemet
- **Ofte ufullstendig og inkonsistent** – ukorrekte mentale modeller blir ofte brukt



### Hva former en mental modell?

Faktorene som former mentale modeller er:

1. **Erfaring fra bruk av systemet (prøving og geiling) eller lignende system**
2. **Metaforer** (eks: skrivebord, billettmaskin, bøker)



3. **Samtaler med andre brukere**
4. **Opplæring og brukermanualer**

### Typer mentale modeller

Vi skiller mellom to typer mentale modeller:

- **Funksjonelle modeller (Hvordan gjør jeg det?)** – basert på **funksjonell forståelse**, der brukeren har sett på hva som må gjøres for å oppnå noe. For eksempel for å handle en billett med AtB Mobillett-appen kan brukeren lage en funksjonell mental modell ved å finne ut hva som må gjøres for å betale billetten, vise den, motta den, osv.
- **Strukturelle modeller (Hvorfor virker det slik?)** – basert på kunnskap om hva som fører til hva, der brukeren har sett på **strukturen til produktet**. For eksempel for et termostat kan brukeren se på den underliggende modellen.

For eksempel kan vi se på en termostat. Den funksjonelle mentale modellen kan bære basert på at termostaten fungerer som en gassgrill, der man skrur med klokken for å øke varmen og mot klokken for å redusere varmen. Den strukturelle modellen kan være basert på kunnskapen om at en termostat kan settes til ønsket temperatur eller at termostaten tilpasser seg til ønsket temperatur.

### Hvordan få tilgang til brukerens mentale modell?

For å få tilgang til brukerens mentale modell kan det utføres:

1. **Brukarhetstester der brukeren bes om å «tenke høyt»**
2. **Intervjuer**
3. **Oppgaveanalyse sammen med brukeren**

Hvis mange brukere gjør den samme «feilen» er det ofte fordi de har uegnet mental modell, og for disse brukerne vil brukskvaliteten til systemet reduseres. **Hvis den mentale modellen ikke korresponderer med den konseptuelle modellen, vil systemet ha få dårlig brukskvalitet.** For å unngå dette må brukernes mentale modell forbedres ved å for eksempel gi bedre opplæring og verktøytips, eller systemet må tilpasses brukernes mentale modell ved å for eksempel flytte elementer dit brukerne forventer å finne dem, endre metaforer og endre konseptuell modell.

#### 3.3.1 Mentale modeller <sup>(B)</sup>

Et suksessfullt system er basert på en konseptuell modell som lar brukere lett lære hvordan de skal bruke systemet effektivt. Brukerens mentale modell er brukerens kunnskap om hvordan systemet fungerer. Den mentale modellen vil utvikles ettersom brukerne lærer mer om systemet. **Det er vanlig at brukere lar ukorrekte mentale modeller lede oppførselen deres**, for eksempel at man trykker flere ganger på knappen i heisen (*more is more*). Dette skyldes tidligere erfaringer med bestemte enheter som for eksempel en kran, der jo mer du dytter på kranen desto mer vann kommer ut. **Mennesker vil utvikle en mental modell om hvordan ting fungerer og vil bruke dette på en rekke enheter uansett om det er passende eller ikke.** Mange mennesker har ufullstendige mentale modeller for bruk av teknologier og tjenester (eks: internett) som er basert på upassende analogier og overtro. Konsekvensen av dette er at de sliter med å identifisere, beskrive eller løse et problem.

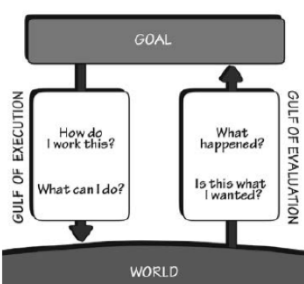
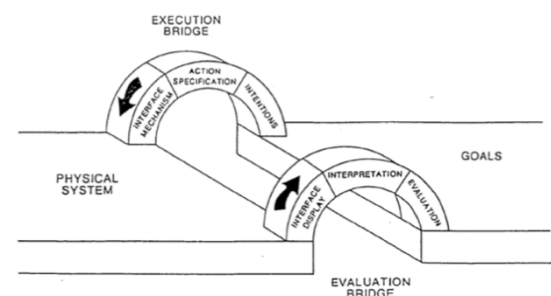
Hvis mennesker kan utvikle bedre mentale modeller for interaktive systemer, vil det bli lettere for de å utføre oppgaver effektivt og vite hva de skal gjøre når det oppstår feil i systemet. **Hvis de klarer å utvikle en mental modell som bedre matcher den konseptuelle modellen vil de få en bedre brukeropplevelse.** For at dette skal bli lettere å oppnå må systemet ha klare instruksjoner som er lette å følge og passende hjelp, opplæring og veiledning som tilpasses brukernes nivå.

## Gulfs of execution and evaluation <sup>(F)</sup>

**Gulf of execution og Gulf of evaluation beskriver gapet som eksisterer mellom brukeren og grensesnittet.** De viser hvordan grensesnittet kan designes slik at brukerne forstår det, altså hvordan gapet kan minimeres.

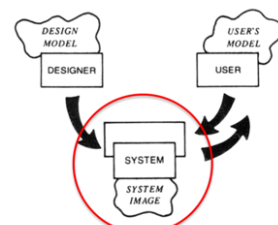
- **Gulf of execution = forskjellen mellom intensjonen til brukeren og hva systemet lar dem gjøre. Det er hvor godt systemet informerer brukerne om hvordan de skal interagere med systemet for å nå målene sine.** Det er differansen mellom hvilke interaksjoner systemet tillater og hva brukeren oppfatter at det er mulig å gjøre med systemet. For eksempel hvis brukeren ønsker å lage en espresso og tror det krever at man trykker på en knapp, men kaffemaskinen krever at man velger dette i en meny og deretter trykker på knappen, vil det være en *gulf of execution*. Vi ser på hvor godt handlingsmulighetene til systemet matcher brukerens tiltenkte handlinger.
- **Gulf of evaluation = forskjellen mellom hvordan systemet informerer om sin tilstand og hvordan brukeren tolker tilstanden. Det er hvor vanskelig det er å oppdage og tolke tilstanden til systemet.** Gulf of evaluation er liten dersom systemet gir informasjon om tilstanden som er lett å få, lett å tolke og matcher måten personen tenker på systemet. Hvis systemet ikke «presenterer seg» på en måte som lar brukeren utlede hvilke handlingssekvenser som vil føre til at målet oppnås, vil det være en *gulf of evaluation*. For eksempel hvis kaffemaskinen går tom for vann og ikke informerer om dette, kan det hende brukeren tror at den ikke tapper kaffe fordi den er tom for kaffebønner, og dermed er det en *gulf of evaluation*.

Gulf of execution vil avhenge av mekanismene som gis av grensesnittet, handlingen som skal utføres og intensjonen til brukeren, mens Gulf of evaluation vil avhenge hva grensesnittet viser og hvordan dette tolkes og evalueres av brukeren. Disse gapene mellom brukerens mål og det fysiske systemet må dekke for å redusere innsatsen som trengs for å utføre en oppgave. Dette må gjøres ved å tilpasse grensesnittet til brukernes egenskaper eller ved å lære brukerne å utføre handlinger som passer med hvordan grensesnittet fungerer.



## Systemmodellen (system image) <sup>(F)</sup>

**System image består av brukergrensesnittet og eventuelle brukermanualer.** Det vil kommunisere den konseptuelle modellen til brukeren, der designmetaforen er spesielt viktig. **Den mentale modellen vil styre hvordan brukeren interagerer med system image, samtidig som interaksjonen vil forme den mentale modellen.**



# F42 – Designmetaforer

Læringsmål:

- Forstå hva en designmetafor er, og dens funksjon i et grafisk brukergrensesnitt
- Kjenne til fordeler og ulemper tilknyttet bruk av metaforer i brukergrensesnitt
- Vite om aspekter det kan være lurt å tenke gjennom i forbindelse med valg av metafor

## Metaforer (F)

Det meste av vår virkelighetsoppfattelse, tanker og språk er bygd opp av metaforer, og de er direkte knyttet til persepsjon og forståelse av verden. For eksempel kan vi se på tid som er et abstrakt begrep som brukes for å betegne hendelsers konstante bevegelse fra fortid til nåtid til fremtid.

Det er vanlig å oppfatte og forstå tid vha metaforer som for

eksempel en klokke, timeglass, kalender, solur eller tidslinje. Vi bruker også metaforer i måten vi snakker om tid på, for eksempel kan vi si at vi sparer tid, bruker tid, har for mye eller lite tid, osv. Vi lever på lånt tid, tiden blir spist opp eller vi kan sammenligne tid med penger, vann eller sand. **Metaforer er kraftfulle semantiske verktøy som brukes for å overføre både overfladiske og dype likheter mellom kjente og nye situasjoner.**



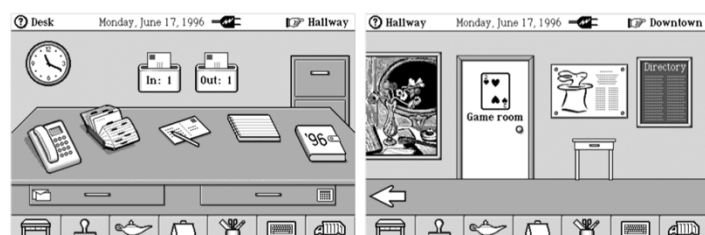
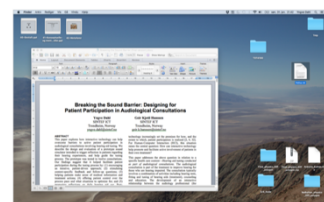
## Rollen til metaforer i GUI-design

**Metaforer spiller en viktig rolle i å kommunisere den konseptuelle modellen til brukerne og i formingen av brukerens mentale modell.** De beskriver hvordan systemet fungerer ved å bruke likhetene det har til et annet mer kjent system. Fordelen med metaforer er at det kan gjøre det lettere for brukerne å danne seg en egnet mental modell av systemet de bruker, dvs. det blir **lettere å lage en mental modell som likner tilstrekkelig på den konseptuelle modellen.** Ulempen er at metaforer fremhever enkelte egenskaper og undertrykker andre, ved at metaforen «lyser opp» og overkjører andre deler av objektet.

**Metaforer vil ofte være en sammensmelting av ulike konsepter,** for eksempel vil skrivebordsmetaforen være sammensatt av dokumenter, mapper, vinduer, søppelbøtte, menyer, osv. **Brukere vil ofte benytte romlige metaforer for å beskrive informasjon,** for eksempel at «det er ikke *her*» eller «det var *i* den delen av databasen». Grensesnittet bør også bruke romlige metaforer i kommandoer og menyer.

## Dårlige metaforer

**Kjennetegn ved dårlige metaforer er at de er unødvendig komplekse og ineffektive.** Et eksempel er Magic Cap, der metaforen forsøkte å gjenskape strukturen i den fysiske verden ved at brukeren måtte flytte seg mellom ulike rom for å få tilgang til ulike ressurser (epost, spill, osv.).



## Valg av metaforer

**Vellykkede designmetaforer bruker metaforer som samsvarer med brukerens konseptuelle oppgaver**, for eksempel kan skrivebordsmetafor brukes for kontorarbeid, mens tengemetafor brukes for designere. Metaforen samsvarer altså med oppgavene som den konseptuelle modellen skal løse. **Metaforen vi velger bør representere systemets virkemåte**, for eksempel kan papirbillett brukes som metafor for AtB Mobillett-appen.



**Metaforer kan gå ut på dato**, for eksempel blir diskettsymbolet brukt som metafor for lagring selv om de fleste ungdommene ikke vet hva en diskett er for noe. Det er viktig at metaforen ikke sender ut feil følelsesmessig signaler og man bør vurdere om metaforen egner seg for brukerne, for eksempel om systemet skal brukes av barn.

### 2.4 Grensesnittmetaforer (B)

Metaforer er en sentral del av konseptuelle modeller, og de gir en struktur som har like aspekter som et annet system og i tillegg egen oppførsel og egenskaper. Metaforen brukes for å fremheve essensen ved en prosess ved å koble den til aspekter ved et annet kjent system. **Målet med metaforer er at de skal gjøre det lettere for brukeren å forstå hvordan systemet fungerer, altså øke brukskvaliteten ved å gjøre det lettere å matche brukerens mentale modell med den konseptuelle modellen.**

**Grensesnittmetaforer kobler kjent kunnskap med ny kunnskap på en måte som gjør det lettere for brukerne å forstå produktet. Valg av metaforer vil være en balanse mellom det som er brukbart og gøyalt, og det er basert på en forståelse av brukerne og brukskonteksten.** For eksempel kan vi se på et utdanningssystem som skal lære 6-åringer matte. En mulig metafor vil være et klasserom med en lærer som står ved tavlen, men hvis man ser på brukerne av systemet vil det være mer sannsynlig at de blir motiverte av noe mer gøyalt som for eksempel et sirkus, ballspill, osv. For å lage gode metaforer må vi:

1. Forstå hva systemet skal gjøre
2. Forstå hvilke deler av produktet som sannsynligvis vil forårsake problemer.  
Metaforer er delvis mapping og bør brukes på områder der brukerne trenger hjelp
3. Lag metaforer ved å se på brukernes oppgaver eller domenet til applikasjonen

#### Kjennetegn ved en god metafor:

- Den gir **kjent struktur**
- Den er **relevant for problemet**, slik at man unngår at brukerne tror de forstår mer enn det de egentlig gjør (fører til falske forventninger)
- Den er **lett å representere** vha visuelle og audio elementer
- Den er **lett å forstå** av brukerne
- Den **kan utvides** ved behov



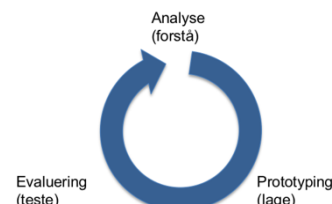
# F50 – Brukersentrert design

Læringsmål:

- Få en bedre innsikt i den brukersentrerte designprosessen og stegene som inngår
- Få kjennskap til sentrale UX-aktiviteter og -leveranser
- Få kjennskap til styrker og svakheter ved de ulike metodene som brukes i brukersentrert design

## Brukersentrert design <sup>(F)</sup>

Brukersentrert design er en **iterativ prosess der brukerne blir aktivt involvert**. Det er sterkt fokus på at **utviklerne må forstå og kunne formulere brukskonteksten**, altså hvem brukerne er, hva produktet skal brukes til og hvor og i hvilken sammenheng det skal brukes. **Konkrete brukes for å kommunisere med brukerne, for eksempel prototyper (uferdige løsningsforslag), scenarier og personas**. Dette er fysiske gjenstander som man kan manipulere. Resultatet ved de ulike fasene i en brukersentrert designprosess er:



1. **Spesifisering av brukskontekst:** beskrivelse av brukskontekst
2. **Spesifisering av brukerkrav:** beskrivelse av brukskontekst, brukernes behov og brukerkrav
3. **Produksjon av designløsning:** spesifisering av brukerinteraksjoner og brukergrensesnitt, og implementert brukergrensesnitt.
4. **Evaluering av designløsning:** evalueringresultat, avvik og langtids overvåkingsresultat

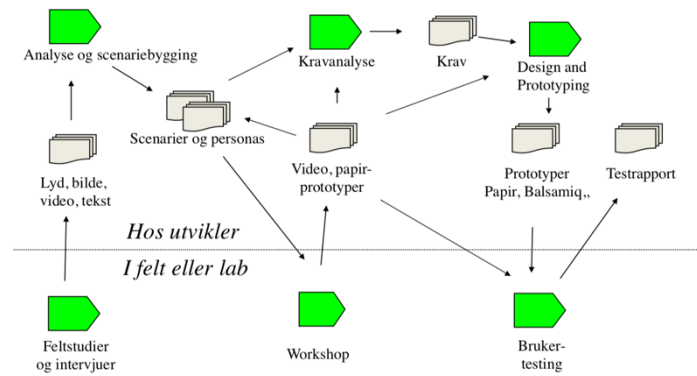
## UX aktiviteter og UX leveranser

**Brukersentrert design er en iterativ prosess som omfatter flere UX-aktiviteter (brukersentrerte metoder og teknikker), som resulterer i ulike UX-leveranser.** UX aktiviteter og -leveranser er ikke nødvendigvis bundet til brukersentrert design, for de kan også brukes i *agile* og *lean* utvikling eller evaluering av eksisterende produkter (eks: hvilken løsning av flere som har best brukskvalitet).

Noen UX aktiviteter er bruksintervju, feltobservasjoner, workshops, spørreundersøkelser, prototyping, brukertesting i lav og felt, posttestintervju, posttest spørre undersøkelser og logging av bruk. Noen UX leveranser er beskrivelse av brukskontekst, brukerkravspesifikasjon, personas, brukerscenarier, konseptuell modell, prototype, designløsning eller testrapport. Figurene under viser hvilke UX aktiviteter som kan gi ulike UX leveranser



Figuren viser **sammenhengen mellom noen UX-aktiviteter og leveranser**. I felt eller på lab kan det gjennomføres feltstudier, intervjuer, workshop og brukertesting, mens hos utvikleren kan det gjennomføres analyse og scenariebygging, kravanalyse og design og prototyping. Noen UX-aktiviteter vil utgjøre en kjede av aktiviteter som er nødvendig for å produsere en bestemt UX-leveranse. For eksempel vil testrapporten involvere flere aktiviteter og andre UX-leveranser.



Fase i brukssentrert design	UX-aktivitet	UX-leveranse
<b>Spesifisering av brukskontekst</b>	<ul style="list-style-type: none"> <li>Feltstudier og intervjuer</li> <li>Workshops</li> <li>Logging av brukermønstre (eksisterende system)</li> </ul>	<ul style="list-style-type: none"> <li>Scenarier og personas (dagens situasjon)</li> <li>Service blueprints</li> <li>Loggdata-analyse</li> </ul>
<b>Spesifisering av brukerkrav</b>	<ul style="list-style-type: none"> <li>Intervju</li> <li>Workshops</li> </ul>	<ul style="list-style-type: none"> <li>Scenarier og personas (tenkt system i bruk)</li> <li>Brukerkravspesifikasjon</li> </ul>
<b>Produsering av designløsning</b>	<ul style="list-style-type: none"> <li>Prototyping (ikke-funksjonell og funksjonell)</li> <li>Scenarier med personas (løsning i bruk)</li> </ul>	<ul style="list-style-type: none"> <li>Scenarier og personas (visning)</li> <li>Prototyper (demo)</li> </ul>
<b>Evaluering av designløsning</b>	<ul style="list-style-type: none"> <li>Bruker-testing (lab/felt)</li> <li>Fokusgrupper for feedback</li> <li>Logging av bruk</li> <li>Spørreskjema (SUS)</li> <li>Heuristisk evaluering (designprinsipper)</li> </ul>	<ul style="list-style-type: none"> <li>Testrapport</li> <li>Oppsummering av feedback</li> <li>Analyser av felt-tester og loggdata</li> </ul>

I fase to blir brukerkravene spesifisert, og figuren viser et eksempel på en brukerkravspesifikasjon. Brukerkravene bør være:

- **Verifiserbare** – «Systemet må være brukervennlig» erstattes med «Brukergrensesnittet må være menydrivet og tilby dialogbokser, muligheter for hjelpe og elementer for relevant brukerinntak» for det er lettere å verifisere
- **Klare og konsise** – «Navigering mellom skjermbilder må skje raskt» erstattes med «Når en bruker navigerer til et nytt skjermbilde må det vises på monitoren innen to sekunder» for det er mer konsist
- **Fullstendige** – «Ved tap av strømtilførsel må vanlige operasjoner kunne utføres vha batteri-backup» erstattes med «Ved tap av strømtilførsel må vanlige operasjoner kunne utføres i minimum 20 minutter vha batteri-backup» for det er mer fullstendig
- **Sporbare**
- **Oppnåelig/realistisk**
- **Nødvendig**
- **Implementasjonsfri**

ID	Krav	Må	Bør	Kan	Kommentarer/forklaring	Kriterier
1.	Applikasjonen må gi brukeren mulighet til å rapportere sin subjektive oppfattelse av dagsform kontinuerlig og på en entydig måte.	1				
2.	Applikasjonen må gi brukeren en kvittering (verifikasjon) hver gang rapporter om hans/hennes dagsform mottas av Helsevakta	1			Viktig at tidspunkt for siste rapport er lett tilgjengelig vi brukergrensesnittet. Historikk kan fungere som referansepunkter	
3.	Applikasjonen bør gi brukeren en oversikt over tidligere innrapportert data om dagsform.		2			

## Egenskaper ved UX-aktiviteter

**Alle brukersentrerte metoder har svakheter og potensielle styrker, og designere må ofte balansere disse.** For eksempel må de balansere kontroll-realisme og kvalitet-kvantitet. Det er ingen metoder som oppfyller alle ønskelige kriterier, fordi alle metodene har ulike svakheter man ikke kan unngå. **Brukersentrert design vil derfor innebære bruk av flere metoder på ulike steg i den iterative prosessen.**

## Brukertesting på lab vs. i felt

For eksempel kan vi sammenligne brukertesting på lab og i felt:

- **Fordeler med labtesting (kontroll)** = brukertester som utføres på lab gir mer kontroll, men mindre realisme. Fordelen med høy kontroll er at man kan si med sikkerhet at resultatet skyldes en bestemt del av brukergrensesnittet. Laboratoriet har et rikt sett med utstyr for datainnsamling (kamera, mikrofoner, osv.) og opptaksutstyret er klargjort for bruk og optimalisert plassert. Brukertesting på lab gir umiddelbar aksess til relevante brukssituasjoner og lignende situasjon kan utspilles flere ganger. Komplexiteten kan reduseres slik at man får data om det man er interessert i og ingen eksterne personer eller hendelser vil påvirke datainnsamlingen.
- **Fordeler med felttesting (realisme)** = brukertester som utføres i feltet gir mindre kontroll, men mer realisme. Brukertesting i feltet gjør at brukeren blir studert i den reelle brukskonteksten og man kan identifisere «uventede hendelser» som påvirker bruken og brukskvaliteten, for eksempel snø, sollys, kulde, påkledning, osv. Det kan gi en bedre innsikt i hvordan designløsningen passer inn i brukerens daglige liv og påvirker andre interessenter (kolleger, pårørende, venner, osv.).

Brukertesting på lab gir altså kontroll på alle parametere unntatt realismen og det har ofte fokus på målbarhet/kvantitet, repeterbarhet og objektivitet.

Brukertesting i feltet studerer bruk i den reelle brukskonteksten og det har ofte fokus på kvalitet (hva betyr dette for brukerne).



## Kvalitativt vs. kvantitativt

Brukertesting på lab har ofte fokus på kvantitative mål, mens brukertesting i feltet har ofte fokus på kvalitative mål. Noen eksempler er:

- **Kvantitative** = tid for å utføre en oppgave eller antall utførte oppgaver
- **Kvalitative** = brukerens beskrivelse av en opplevd utfordring med systemet eller artefakter som blir laget i forbindelse med en observert aktivitet (eks: video)



## Andre UX-aktiviteter

Tabellen viser formal, type måling, styrker og svakheter ved ulike UX-aktiviteter.

Metode	Formål	Type data	Styrker	Svakheter
Intervju/fokusgruppe	Utforske, få innblikk i tema	Kvalitativt	Presisjon	Respons uavhengig av brukerkontekst
Workshop	Dialog med brukere, Muliggjør samskaping	Kvalitativt	Diskusjon, drøfting av et tema; En trenger ikke lage detaljerte spørsmål i forkant; Produserte artefakter	Ofte kontekst uavhengig Krever ofte tidkrevende analyse (koding) av data
Felt-observasjon	Studie av reell brukerkontekst	Kvalitative	Realisme, Liten påvirkning på det som observeres	Mindre kontroll, Tidkrevende, Vanskelig datainnsamling
Brukertesting (lab)	Evaluere brukskvalitet	Kvalitativ og kvantitativ	Kontroll; Detaljerte opptak; Rask aksess til relevant situasjon	Manglende realisme;
Brukertesting (felt)	Evaluering brukskvalitet	Kvalitativ	Realisme	Mindre kontroll, Tidkrevende, Vanskelig datainnsamling
Spørreskjema	Få svar på konkrete spørsmål	Kvantitativ	Kan nå ut til mange, (generaliserbarhet)	Vi må stille de «riktige» spørsmålene; Kontekst tas ikke i betraktning.
Logging av bruk	Spore faktisk bruk	Kvantitativ	Kan finne mønster i bruk (vanskelig å se).	Det kan være vanskelig å se sammenheng mellom log og brukerkontekst.

Merk: brukersentrert design vil innebære bruk av flere av disse metodene ved ulike steg i den iterative prosessen. Dette brukes for å få kunnskap om brukerne og brukerkonteksten, slik at brukerne blir involvert i designet. Målet er at brukeropplevelse skal bli bedre!

Vi ser nærmere på noen av UX-aktivitetene:

- **Intervju** – brukes for å utforske og få innblikk i temaet og kan utføres individuelt eller i gruppe. Et viktig valg er hvem man skal intervju. Intervju kan være strukturert, semi-strukturert eller åpent. Det blir ofte brukt en intervjuguide som leder intervjuet gjennom ulike steg. Etter intervjuet må dataen analyseres. Styrken ved intervju er at det kan være presist, mens svakheten er at responsen er uavhengig av brukskonteksten.
- **Workshops** – brukes for å ha en dialog med brukerne og muliggjøre samskapning. Et antall brukere blir samlet for noen timer eller en hel dag, der man diskuterer utfordringer, utforsker designrommet og får feedback på systemet. Det er viktig å ha en detaljert plan klargjort på forhånd og en plan B i nødfall. Styrkene er at man får drøftet et tema, trenger ikke detaljerte spørsmål og kan produsere artefakter (eks: video og lydopptak), mens svakhetene er at responsen ofte er uavhengig av brukskontekst og det krever mye analyse av dataen.
- **Spørreskjema** – brukes for å få svar på konkrete spørsmål og er dermed en kvantitativ måling. Man må identifisere brukergruppen, lage et godt spørreskjema og bruke tid på å analysere resultatet. Det kan være lurt å gjøre noen intervju og observasjoner på forhånd for å unngå hull i spørreskjemaet. Styrken er at man kan nå ut til mange, mens svakhetene er at man må stille riktige spørsmål og responsen er uavhengig av brukskonteksten.
- **Logging av bruk** – brukes for å spore faktisk bruk og gir kvantitative mål. Det krever en installasjon av automatisk logging og vil generere anonymiserte loggdata. Dette kan gi viktig feedback på en tidlig versjon av et system, men det krever analyseverktøy for å se sammenhenger. Styrken er at man kan finne bruksmønstre som ellers er vanskelig å observere, mens svakheten er at det kan være vanskelig å se sammenhengen mellom loggdataen og brukskonteksten.





# F51 – Personas og scenarier



Læringsmål:

- Vite hva en personas er og dens rolle som verktøy i brukersentrert design
- Forstå hva som er hensiktsmessig at inngår og ikke inngår i personasbeskrivelse
- Forstå rollen til scenarier som verktøy i brukersentrert design, i form av kommunikasjon, ulike typer, samspill med personas og abstraksjonsnivå

Forhold mellom brukskvalitet, personas og scenarier

Brukskvalitet er i hvilken grad produktet er anvendbart, effektivt og tilfredsstillende ved:

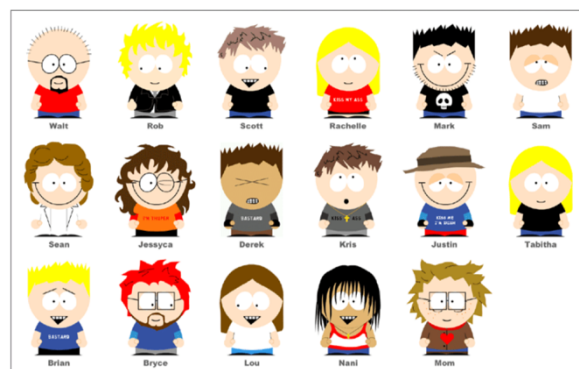
- Bestemte brukere
- Bestemte mål
- Bestemte omgivelser (brukskontekst)

Personas brukes for å beskrive de bestemte brukerne, mens scenarier brukes for å beskrive de bestemte målene og omgivelsene (brukskonteksten). I eksempelet med bestikk på tur, i middagselskap eller på asiatisk restaurant, vil personas være de som skal spise, mens scenarie er hvordan de skal spise og hvor de spiser.



Personas (F)

Personas er brukerkarakterer og ikke virkelige personer, og de brukes for å representere ulike brukersegmenter. Slike brukersegment kan for eksempel være studenter, unge i jobb, pensjonerte, osv. Som regel vil 3-5 personas være tilstrekkelig for å dekke alle segmentene. Personas brukes for å hjelpe designere med å bli konkrete i forhold til designløsningen. For eksempel kan de tenke at de designer produktet for brukerkarakteren Espen (25) istedenfor «de yngre brukerne».



En personasbeskrivelse vil inkludere aspekter som er relevant i forhold til designløsningen, og inkluderer mål, motivasjon, egenskaper og oppførsel. Den generelle strukturen kan være:

1. Navn, alder, jobb, sivil status, boform
2. Litt om livet (interesser, personlighet, kjerneverdier)
3. Beskrivelse av tilhørende brukersegment (statistikk)

## Espen - Ung i jobb 5%

- 25 år
- Single, bor i kollektiv
- Internettredaktør

• Espen bor sammen med to kompiser i leilighet på Sagene. Han jobber i Internettredaksjonen til Dagsavisen. Han spiller basketball og er opptatt av film. Espen har bare vært Telenor-kunde i ett år. Tidligere var han sene kunde pga prisen.

• Espen er veldig sosial, og bruker telefonen mye til å koordinere fritidsaktiviteter med venner. Han ringer for å avtale ting, eller sender sms underveis. Mobilten er alltid på. Han får ikke dekket mobilutgiftene på jobben. I det siste har Espen fått øynene opp for mobil internett. Han liker spesielt nyheter.

**Kjerneverdier**

"No problem"  
Ubekymret – Mer penge å rutte med  
Sport & Spill

**Beskrivende karakteristika**

- 50/50 kvinner og menn
- 53% er i jobb, 37% student/skoleelev
- 1 av 3 samboere, 2 av tre enslige
- 25% i T10

**Verdier og holdninger til bruk**

- Tilgjengelighet for kunder og kolleger viktig for over halvparten
- Kostnader relevant – 39% vil bytte leverandør hvis de kan tjene på det, men 57% mener funksjoner er viktigere enn pris

**Adferd**

- Middels internettbrukere
- 93% betaler selv og 3% får dekket av arbeidsgiver

**2,5 – 3G / mCommercesprodukter**

- Mange ser fram til mobil internett (49%)
- Tilgang til jobbsystemer og kalenderfunksjoner for 1 av 3
- Fjernstyring av elektriske installasjoner aktuelt for 1 av 2

**Ekstisterende kundeløsninger**

- Fløst med Ring Kontakt (45%), 42% har Privat og 13% Finnvar

## Jill Anderson

**Bio**

Jill is a Regional Director who travels 4-8 times each month for work. She has a specific region in which she travels, and she often visits the same cities and stays in the same hotel. She is frustrated by the fact that no matter how frequently she takes similar trips, she spends hours of her day booking travel. She expects her travel solutions to be as organized as she is.

**Personality**

Extrovert	Introvert
Sensitiv	Intuisjon
Thinking	Feeling
Judging	Perceiving

**Brands**

KAYAK Basecamp Outlook enterprise IHG

**Goals**

- To spend less time booking travel
- To maximize her loyalty points and rewards
- To narrow her options when it comes to shop

**Frustrations**

- Too much time spent booking - she's busy!
- Too many websites visited per trip
- Not terribly tech savvy - doesn't like the process

**Motivations**

Price  
Comfort  
Convenience  
Speed  
Preferences/Rewards  
Loyalties/Rewards

**Technology**

IT & Internet  
Software  
Mobile Apps  
Social Networks

**"I'm looking for a site that will simplify the planning of my business trips."**


AGE 29  
OCCUPATION Regional Director  
STATUS Single  
LOCATION Portsmouth, NH  
TIER Frequent Traveler  
ARCHETYPE The Planner

Organized	Practical
Protective	Hardworking

## Kivio Users


	The researcher	The Sysadmin	The OSS developer	The CS student
Name	Alexander Weiß	Donald M. Berry	Kristian Larsson	Eric Neville
Age	30	30	26	24
Location	Germany	US	Sweden	France
Social Life	Alexander lives with his girl-friend in a flat in Hamburg.	Donald lives with his wife and 1-year old daughter in a house in Portland.	Kristian shares an apartment with two friends in Stockholm. His girl-friend lives in Uppsala. They see each other every weekend.	Eric lives with his parents in a small city close to Lyon. He visits the university there. Often, he stays at his friend's apartment for playing PC games and programming.
Work Life	He works at centre for environmental systems research and designs plans for replacable energies in a EU-funded project.	He is a lead system administrator in a huge network solutions company in Portland.	A software developer with a dayjob in a medium-sized software company. Works on KDE in his spare time.	He is a student of computer science. Besides university, he performs small programming jobs for people in his neighbourhood.
Computer Experience	All are highly experienced with computers.			
Time at a computer per week	26-50 hours per week	35-50++ hours per week	30-50++ hours per week	25-45 hours per week
Computer tasks	Office tasks and Field-dependent. Also educational and recreational. No development.	Development and network administration. Does not use PC for office tasks, educational, and even recreational.	Mostly development and recreational. Also network administration and office.	Mostly development. Also educational, recreational, and network administration. Does not use for office work.
Relation to OSS.	He is not passionate about OSS.	He is a convinced user of OSS.	He is involved with OSS development.	He is a convinced user of OSS.
Requirements wrt diagramming	office requirements	highest claims	easy-going	eager beaver
Frequency of drawing diagrams	Each 2 <sup>nd</sup> month	Twice per month	Once a month	Each 2 <sup>nd</sup> month

## PERSONAS




**Iggy the Influencer**  
Age: 36-45  
Gender: M  
Marital Status: Married  
Education: Bachelors  
Readability: Grade 18+  
Career Type: Freelance  
Career Stage: C-Level or higher  
Income Range: \$41k-\$70k

Iggy is an influencer who lives online, no decision is made in his life without first checking the web, and thus he has earned a very strong following in the content marketing industry. He works as a freelancer but has a background in web development, SEO, PR and digital marketing.




**Wendy the Writer**  
Age: 26-35  
Gender: F  
Marital Status: Single  
Education: Some College  
Readability: Grade 18+  
Career Type: Steadily Employed  
Career Stage: Entry\_Level  
Income Range: \$41k-\$70k

Wendy is a freelance writer and content curator. She has a base knowledge of content marketing, but identifies as a creative writer first. She runs her own blog and is always looking for actionable tips on how to increase engagement on her blog and through social media, tips and tools to manage work flow and SEO.



**Marketing Mark**  
Age: 26-35  
Gender: M  
Marital Status: Single  
Education: Bachelors  
Readability: Grade 13-15  
Career Type: Steadily Employed  
Career Stage: Senior Management  
Income Range: \$71k-100k

Mark is a marketer and is always looking for trending content and industry news. Mark is connected to the internet all day, every day, so to get his attention, you either need brand news, actionable information or a completely unique approach. Mark loves workflow-tool pieces and data backed case studies.



**CEO Cecilia**  
Age: 26-35  
Gender: F  
Marital Status: Single  
Education: Bachelors  
Readability: Grade 16-18  
Career Type: Steadily Employed  
Career Stage: Senior Management  
Income Range: \$71k-100k

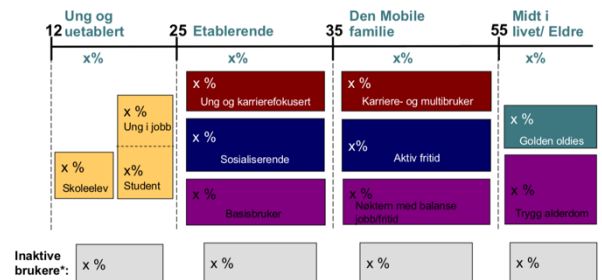
Cecilia runs a big company and is on the lookout for ways for her company to run more efficiently. She gravitates towards articles that offer ways to simplify your workflow, basic management tips, and tools to improve day-to-day efficiency. Cecilia always keeps ROI in mind and articles she reads do too.

Personas gjør det lettere å sammenligne egenskaper hos brukerne.

Bruk og produksjon av personas:

1. **Velg målgruppe(r) for produktet**
2. Identifiser brukersegmentene og **representer hvert segment med én brukerkarakter**. Baser brukerkarakteren på virkelige data om segmentet, ikke ren fantasi!
3. **Anvend brukerkarakteren aktivt under designprosessen** og ikke snakk om brukerne generelt. **Vurder om brukerkarakterene ville brukt funksjonene.**

Figuren viser et eksempel på segmentering av mobiltelefonbruk.



Personas kan brukes for å prioritere funksjoner i designet ved å se om tjenesten er nyttig for brukerne eller om de klarer seg godt uten. For å vurdere om brukerkarakterene ville brukt funksjonene kan vi bruke en **personas-vektlagt prioritetsmatrise** (se figur). Hver personas får en vektlegging som avgjør hvor viktig deres mening er, og jo større vekt desto viktigere er meningen. For hver funksjon vil vi summere vektene og **jo større vekt, desto mer nyttig er funksjonen**. Funksjoner som har liten eller negativ vekt bør droppes.

Figure 4: A feature by Persona-weighted priority matrix.

	Persona 1	Persona 2	Persona 3	Weighted Sum
Weight:	50	35	15	
Feature 1	0	1	2	65
Feature 2	2	1	1	150
Feature 3	-1	1	0	-15
Feature 4	1	1	1	100
Etc.	-	-	-	-

-1: Persona irritert over egenskap  
0: Persona bryr seg ikke om den er der eller ikke  
1: Egenskapen vil ha en verdi for personaen  
2: Personaen vil elske denne egenskapen

Hva personasbeskrivelse *ikke* skal inneholde

Noen eksempler på hva som ikke bør inngå i en personasbeskrivelse er:

- **Karikaturer** (komisk fremstilling), **overdrivelser**, **ekstreme eller unaturlige ferdigheter**
- **Holdninger til enkelte brukersegmenter** (eks: dataspill er skadelig)
- **Sterke assosiasjoner til kjente personer** (eks: bilde er sinna-snekkeren eller statsministeren)
- **Irrelevante detaljer** (eks: hårfarge eller opplevde enkelthendelser)
- **Brukerkarakterens holdninger til designløsningen**

## Scenarier <sup>(F)</sup>

Scenarier (tekst og bilder) brukes for å skape en felles forståelse blant ulike aktører om et problem eller en mulig designløsning. Designere bruker scenarier for å lettere kommunisere med brukerne, siden de slipper å bruke IT-faglige abstraksjoner som få brukere forstår (eks: objektorienterte arkitekturer, dataflyt-diagram, osv.). Scenarioene gjør designerne blir mer konkrete, og de blir ofte kombinert med personas. Scenarier vil illustrere brukerrepresentanter (personas), oppgaver og omgivelser, og de er basert på tekst, bilde eller begge deler. Det blir lettere for brukerne å tenke vha konkrete eksempler som gis av scenariet. Scenarier er en historie som viser hvordan produktet kan brukes for å oppnå en oppgave.

Det er viktig å finne passende abstraksjonsnivå, slik at scenariet kan brukes for å skape felles forståelse (likhet med personas). Scenariet må gi nok detaljer til at det kan fungere som en felles referanse for ulike aktører, samtidig som det må unngå detaljer som ikke er relevant på det aktuelle tidspunktet i designprosessen. Som regel vil grove skisser være bedre enn detaljerte fotografier. Figuren under viser scenariet for et system som hjelper brukeren med å finne bensinstasjon.

### Gas Station



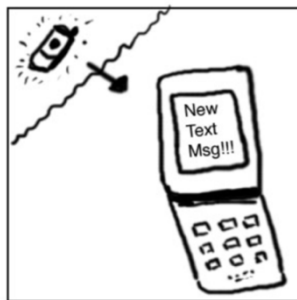
1) User realizes the vehicle is low on gas.



2) Commuter takes out the digital pen and a copy of the local map pre-printed on the Anoto paper to request for information regarding nearby gas stations.



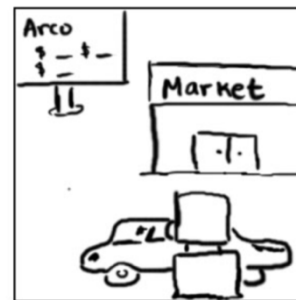
3) Commuter draws a circle where he wants to find gas stations and checks the box under gas.



4) Commuter receives a text message containing information for nearby gas stations.



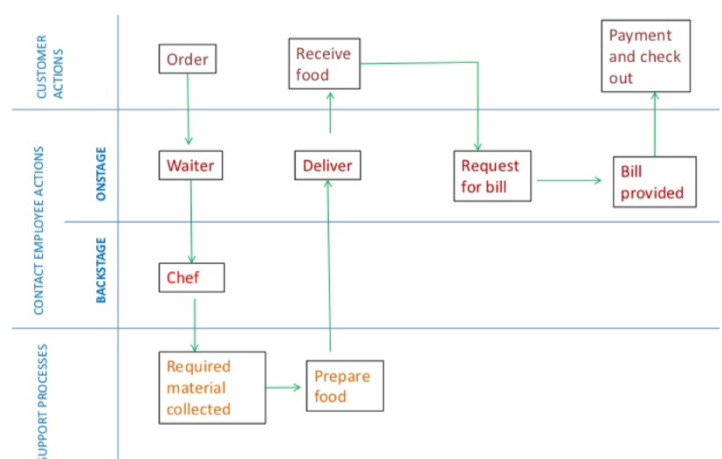
5) The text message will contain the gas station, its current prices, and location.



6) Commuter replenishes the car's supply of gas and avoided being stuck on the road.

## Service blueprints

Et alternativ til scenarier er **service blueprint** (se figur) som er et diagram som viser forholdet mellom ulike komponenter og prosesser som en bruker vil møte på når hen bruker systemet.



## Kapittel 7 – Samling av data <sup>(B)</sup>

Samling av data er essensielt for etablering av brukerkrav og evaluering. Det finnes flere metoder (UX-aktiviteter) for å samle data, og ved design bør man kombinere flere metoder for å unngå skjevheter som de ulike metodene er utsatt for. Det er fem hovedutfordringer ved samling av data:

1. **Målsetting** – det er mange ulike grunner til at man samler data, og før man begynner er det viktig å identifisere målet ved UX-aktiviteten
2. **Identifisering av deltagere** – man må velge hvilke deler av populasjonen som skal delta i datasamlingen (kalles sampling)
3. **Forhold med deltagere** – de som utfører samlingen og de som gir data må ha et klart og profesjonelt forhold
4. **Triangulering** – systemet bør undersøkes fra minst to ulike perspektiv (dvs. flere brukere og flere UX-aktiviteter)
5. **Pilotstudier** – en liten prøvekjøring gjennomføres før hovedstudiet for å sikre at metoden kan brukes og fikse eventuelle problemer

Noen typer UX-aktiviteter:

- **Intervju** = en samtale med et formål. Hvilken type intervju som brukes vil avhenge av målet ved intervjuet, spørsmålene som skal stilles og hvor langt utviklingen har kommet. Intervju brukes for å utforske problemer. Fire typer intervju:
  - **Ustrukturerte intervju** – intervjueren har lite kontroll over intervjuprosessen, som ligner mer på en samtale rundt et spesifikt emne med åpne spørsmål. Brukes ofte for å samle data om brukernes førsteinntrykk av produktet
  - **Strukturert intervju** – intervjueren stiller forhåndsbestemte spørsmål som ofte er lukket (dvs. krever svar fra et sett med alternativer). Kan brukes for å få feedback om bestemte deler av systemet
  - **Semi-strukturert intervju** – bruker både åpne og lukkede spørsmål og intervjueren følger et grunnleggende skript.
  - **Gruppeintervju (fokusgruppe)** – fokusgruppe er en type gruppeintervju der 3-10 brukere er involvert og diskusjonen ledes av en tilrettelegger. Brukes for å samle flere synspunkt.
- **Spørreskjema** = brukes for å få svar på spesifikke spørsmål fra et stort antall deltakere. Det kan inkludere mennesker som er spredt over større områder. Det er viktig at spørsmålene er klart formulert slik at dataen kan analyseres effektivt. Intervju med en liten gruppe deltagere kan brukes for å finne informasjon som deretter kan bekreftes med spørreskjema som sendes til mange deltagere. Spørreskjema brukes når deltakerne er motiverte til å svare på spørsmålene, mens intervju brukes når de trenger overtalelse.
- **Observasjoner** = observasjon kan brukes tidlig i designprosessen for å forstå brukerkonteksten, oppgaver og mål eller senere for å evaluere hvor godt produktet støtter disse oppgavene og målene. Brukerne kan studeres direkte mens de utfører aktivitetene eller indirekte gjennom ulike opptak. Observasjoner kan foregå i feltet (realistisk) eller på en lab (kontrollert):
  - **Observasjoner i feltet** – brukes for å **forstå brukskonteksten**. Det er vanskelig for brukere å forklare nøyaktig hvordan de oppnår en oppgave, så observasjoner i feltet er nødvendig for å fylle inn detaljer om hvordan

brukerne oppfører seg og bruker produktet. Det kan være vanskelig å utføre observasjoner i feltet og det kan føre til at mye data må analyseres.

- **Observasjoner på labben** – vil være mer formelle enn observasjoner i feltet og brukeren vil sannsynligvis være mer stresset. **På labben vil man fokusere på detaljene i det brukeren gjør, mens i feltet vil brukskonteksten være viktig og hvordan brukerne interagerer med hverandre, produktet og omgivelsene.**
- **Indirekte observasjoner** – observasjoner kan være forstyrrende og krever at observatørene er tilstede hele tiden. Alternativet er indirekte observasjoner der dagbøker eller interaksjonslogger brukes for å registrere brukernes aktiviteter i en logg som undersøkes i etterkant.

## Kapittel 9 – Prosessen av interaksjonsdesign <sup>(B)</sup>

I brukssentrert design blir brukerne involvert, og fordeler ved dette er:

- Sikrer at utviklingen fokuserer på brukernes aktiviteter, slik at designerne får en bedre forståelse over brukernes mål og dermed vil produktet bli mer brukbart
- Kontrollerer hva brukerne forventer av produktet, siden de får se hva produktet kan fra starten av
- Gir brukerne en større følelse av eierskap til produktet, siden de er involvert i utviklingen.

I en brukersentrert designprosess vil ekte brukere og deres mål være drivkraften bak produktutviklingen. Grunnlaget for en brukersentrert tilnærming er:

- **Tidlig fokus på brukere og deres oppgaver** – brukerens oppgaver og mål er drivkraften bak utviklingen. Oppførselen til brukerne og brukskonteksten studeres og systemet designes for å støtte disse. Det er viktig å observere hvordan brukerne utfører oppgaver fordi dette vil fremheve preferanser, prioriteter og implisitte intensjoner hos brukerne. Brukere liker ikke å avvike fra deres lærte vaner. Produktene bør ta hensyn til brukernes begrensninger, både kognitive og fysiske, slik at antall feil brukerne gjør blir minimert. Designere må tenke på brukskonteksten og brukerne når de tar designvalg.
- **Empiriske målinger** – målinger og observasjoner av prestasjonen og reaksjonen til brukere som interagerer med scenarioer, prototyper, simuleringer, osv.
- **Iterativt design** – problemer identifiseres og fikses, før prosessen gjentas. Sykluser med design-test-mål-redesign blir gjentatt så ofte som nødvendig. Iterasjon lar designet forfines basert på feedback.

## Kapittel 10 – Etablering av krav <sup>(B)</sup>

Etablering av krav er en iterativ aktivitet, siden kravene vil utvikle seg etterhvert som interessenter interagerer med designet og ser hva som er mulig. Det er dyrt å fikse feil senere i utviklingen, så derfor er det lurt å bruke mye tid på kravspesifikasjon tidlig i designprosessen. Det krever en forståelse av brukerens behov for å kunne identifisere og etablere krav. De bør være så spesifikke, entydige og klare som mulig. For eksempel for en smartklokke GPS kan kravet være at det tar mindre enn et halvt sekund å laste kartet. Designere må kunne avgjøre når kravet er oppfylt. For eksempel hvis kravet er at tenåringsjenter skal være fornøyde med klokken, må designerne undersøke hva som gjør at de blir fornøyde (kravet må bli mer spesifikt).

## Etablering av krav vha datasamling

**Målet ved datainnsamlingen i fasen som etablerer brukerkrav er å samle tilstrekkelig, relevant og passende data, slik at et sett med stabile krav kan produseres. Selv om et initialt sett med krav eksisterer, må datasamling gjennomføres for å utvide, spesifisere og bekrefte disse kravene.** Valg av metode for datainnsamling vil avhenge av egenskapene til oppgavene, deltakerne og ressursene som er tilgjengelige. Det er vanlig at flere metoder brukes for å oppnå ulike perspektiv.

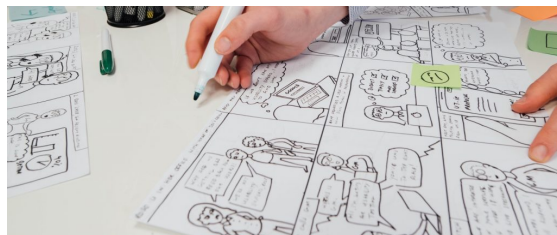
## Personas

**Egenskaper ved brukerne vil påvirke designet til produktet**, for eksempel evner, nasjonalitet, utdanningsbakgrunn, preferanser, fysiske og mentale evner, osv. Designet vil også påvirkes av om det er nye brukere, standard eller eksperter som skal bruke systemet. **Samlingen av egenskaper til en bruker kalles brukerprofil, og brukerprofiler representeres som personas. Personas er rike beskrivelser av typiske brukere av systemet og produktet designes for disse.** En personas vil ikke beskrive en ekte person, men representerer i stedet en bestemt del av brukerne som har felles mål for produktet. Personas vil også inkludere en beskrivelse av egenskaper, holdninger, oppgaver og omgivelser.

Hver personas har et navn, ofte et bilde og noen personlige detaljer (eks: hva de gjør på fritiden). Dette gjør at designerne **kan se på personas som ekte potensielle brukere**. Et produkt vil som regel ha 3-5 personas. Det er en mye brukt metode, siden det er en lett måte å kommunisere egenskaper og mål hos brukere.

## Scenarier

Scenarier brukes for å beskrive eksisterende eller forutsette oppgaver man kan utføre vha systemet. **Det er en uformell, fortellende beskrivelse om menneskelige aktiviteter eller oppgaver, og det legger til rette for diskusjon av kontekst, behov og krav.** De lages med et språk og bilder som kan forstås av brukere og andre interessenter. Å fortelle historier er en naturlig måte for mennesker å forklare hva de gjør og hvordan de oppnår noe. Fokuset til historiene er ofte hva brukerne prøver å oppnå, altså deres mål.



Detaljnivået til scenariet vil avhenge av hvor i utviklingen man befinner seg. I løpet av etableringen av krav bør scenariene fremheve konteksten, mål for brukskvaliteten og brukeropplevelsen og oppgavene som brukerne skal gjennomføre. **Scenariene blir ofte kombinert med personas for å fremheve brukeropplevelsen.** De vil være personlige og viser kun ett perspektiv. **Scenarier kan brukes for å modellere eksisterende funksjoner eller uttrykke mulige bruksområder for produktet. De brukes også i evalueringen av produktet og for å bygge felles forståelse i teamet.**

Som vi kan se på figuren kan scenariet også være kun tekst (dvs. må ikke inkludere bilder).

### Scenario 3: Hyper-wonderland

This scenario addresses the positive aspects of how a hypermedia solution will work.

The setting is the Lindholm construction site sometime in the future.

Kurt has access to a portable PC. The portables are hooked up to the computer at the site office via a wireless modem connection, through which the supervisors run the hypermedia application.

Action: During inspection of one of the caissons<sup>1</sup> Kurt takes his portable PC, switches it on and places the cursor on the required information. He clicks the mouse button and gets the master file index together with an overview of links. He chooses the links of relevance for the caisson he is inspecting.

Kurt is pleased that he no longer needs to plan his inspections in advance. This is a great help because due to the 'event-driven' nature of inspection, constructors never know where and when an inspection is taking place. Moreover, it has become much easier to keep track of personal notes, reports etc. because they can be entered directly on the spot.

The access via the construction site interface does not force him to deal with complicated keywords either. Instead, he can access the relevant information right away, literally from where he is standing.

A positive side-effect concerns his reachability. As long as he has logged in on the computer, he is within reach of the secretaries and can be contacted when guests arrive or when he is needed somewhere else on the site. Moreover, he can see at a glance where his colleagues are working and get in touch with them when he needs their help or advice.

All in all, Kurt feels that the new computer application has put him more in control of things.

<sup>1</sup> Used in building to hold water back during construction.



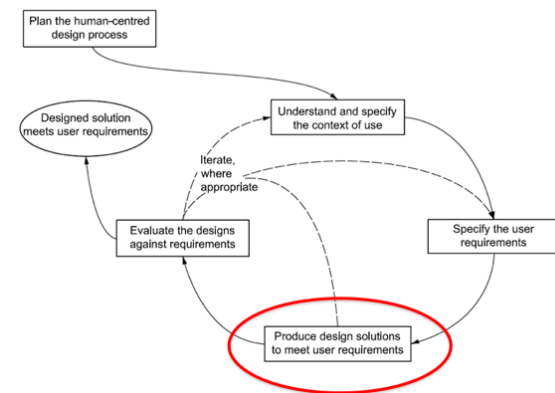
# F60 – Prototyping

Læringsmål:

- Forstå hva en prototype er og dens rolle i interaksjonsdesign:
  - Forstå prinsippet om *just enough* prototyping
  - Få innsikt i ulike former for prototyper

## Prototyper <sup>(F)</sup>

En prototype er en forenklet representasjon av en designløsning og en konkretisering av designideer. Prototyper brukes som et verktøy for testing og gjenstand for tilbakemelding fra brukere. De kan brukes for å vurdere hvor egnet løsningen er for den gitte brukskonteksten. Når brukerne og andre interessenter involveres i designprosessen betyr det at designet og potensielle løsninger må kunne kommuniseres til andre enn designerne. Dette krever at designet kan uttrykkes på en måte som tillater feedback og forbedringer. En måte å gjøre dette på er vha. prototyper. **Formålet til en prototype er å fungere som en gjenstand som brukere kan vurdere og gi tilbakemelding for. Den representerer en designløsning og vil dermed hjelpe til med å drive den brukersentrerte designprosessen fremover.** Hensikten til prototypen er altså ha et materiale som kan testes. Prototyper er en essensiell del av iterativt design.



## Hvorfor lage prototyper

Primære årsaker:

- Kunne empirisk teste en designide, dvs. vurdere den opp mot virkeligheten
- Få innsikt i hvordan designløsningen kan forbedres
- Spare ressurser (tid og penger)

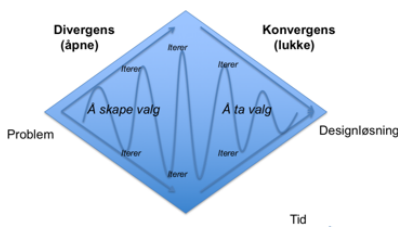
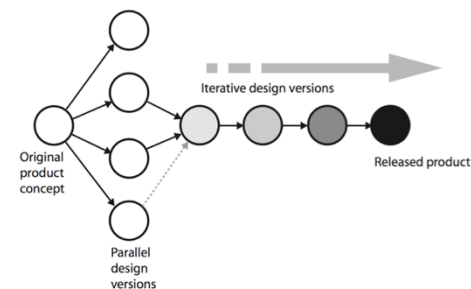
Sekundære årsaker:

- Utforske flere mulige design
- Teste teknisk funksjonalitet
- Kommunisere med brukere, utviklere, oppdragsgivere og andre interessenter

## Livssyklus modell

En livssyklus modell representerer et sett med aktiviteter og hvordan de er relatert til hverandre. For eksempel viser figuren over til høyre den internasjonale standard modellen ISO 9241-210.

Livssyklus modeller er en forenkling av virkeligheten. Figuren til høyre viser **usability engineering livssyklus** laget av Jacob Nielsen. Her kan vi se at det originale produktkonseptet vil resultere i flere versjoner av designet som tilsammen brukes for å produsere en designversjon. Denne vil forfines ved iterasjon til man når det endelige produktet. Dette illustrerer at **interaksjonsdesign er en iterativ prosess med divergens og konvergens.**



## Ulike prototypeteknikker <sup>(F)</sup>

Prototyper lar brukerne evaluere interaktive produkter ved at de får interagere med dem. Det er ulike prototypeteknikker og ikke alle krever fungerende programvare.

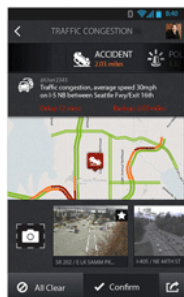
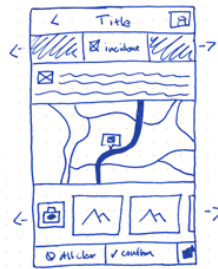
### Just enough prototyping

**Prototyper trenger ikke å være robuste, detaljerte, realistiske, vakre, utvidbare eller komplette. De trenger ikke å overleve lengre enn brukertesten.** Dette er konseptet bak **just enough prototyping** som gir lav investeringskostnad og bygger på at «alle triks er lov!». **Just enough prototyping gir størst avkastning (ny innsikt) i forhold til ressurser (tid, penger) som investeres i prototypen. Minimum viable produkt er et utkast av produktet som er tilstrekkelig nok til at det skaper refleksjon blant potensielle brukere, slik at de kan gi tilbakemeldinger som kan være retningsgivende for fremtidig design.**

### Høynivå og lavnivå prototyper

Vi kan dele prototypene inn i:

- **Lavnivå prototype (low-fidelity, mock-ups)** = enkle (papir)prototyper som har lite grafikk og interaksjon. De ligner ikke mye på det endelige produktet og gir ikke samme funksjonalitet. De er enkle, billige, lette å modifisere og støtter utforskning av alternative design. De brukes tidlig i prosjektet, siden de er enkle å lage, men mangler detaljer. Noen lavnivå prototyper:
  - **Wizard of Oz** = en animert utseendemodell med mennesker og tilgjengelig teknologi «bak kulissene». Brukeren interagerer med prototypen og en menneskelig operator simulerer responsen. Hvis alt fungerer kan det gi en realistisk brukeropplevelse, men det er best egnet for å simulere ett begrenset sett med funksjonalitet.
  - **Storyboard** = brukes ofte sammen med scenarier. Storyboard representerer en sekvens av handlinger som brukeren og produktet går gjennom for å oppnå en oppgave, mens et scenarie er en historie om hvordan produktet kan brukes for å oppnå oppgaven. Scenariet kan deles inn i en rekke steg og storyboard vil da være en scene per steg. Det brukes for at designere skal se mer detaljert på scenariet og bruk av produktet.
- **Høynivå prototype (high-fidelity, funksjonell prototype)** = mer komplekse prototyper med mer detaljer og funksjonalitet. De ligner mer på det endelige produktet. De er mer ressurskrevende å lage og det brukes ofte prototypingsverktøy. De brukes senere i designiterasjoner for å finjustere funksjonaliteten og grafikken. Eksempel er alphaversjon av systemet

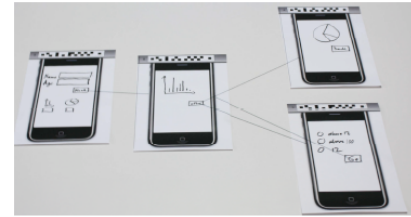
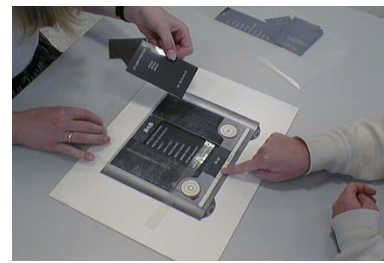


### Papirprototyper (lavnivå prototype)

**Papirprototyper er raske å lage, og de er egnet i tidlige faser av designprosessen.** For eksempel kan de brukes for å sjekke om man har brukt en god konseptuell modell for designet eller for å identifisere potensielle brukbarhetsproblemer før implementasjon. **Papirprototyper blir ofte brukt i brukertester**, fordi de er enkle å endre på under testen (eks: forslag til endringer kan skrives på papiret) og de forenkler kommunikasjonen med testpersonen, siden testpersonen kan vise hva hen mener. En papirprototype kan være frihåndsskisser, utklipp av GUI elementer eller utskrift av skjermbilder laget med Balsamiq, Photoshop, osv.



**Papirprototyper kan kombineres med Wizard of Oz for å simulere interaktivitet med brukeren.** På denne måten kan papirprototypen brukes for å vise funksjonalitet. Den vil samtidig ikke kunne gjenspeile responstid, lyder og animasjon, og det krever at brukeren aksepterer simuleringen som en erstatning for en reell brukssammenheng. I en papirprototype kan **tilstandsdiagram brukes for å vise sammenhengen mellom de ulike skjermbildene:**

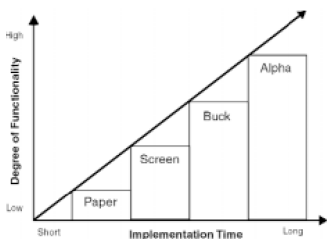
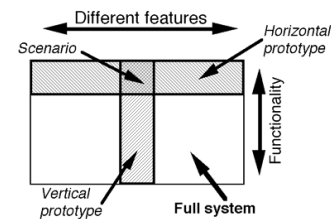


### Kompromisser i prototyping

**Intensjonen med prototyper er å produsere noe raskt som kan brukes for å teste deler av produktet, og dette krever kompromiss.** Det er begrenset hvilke spørsmål en prototype kan svare på, så prototypen må designes med utgangspunkt i hovedmålene. I lavnivå prototyper er det klart at kompromisser har blitt gjort, for eksempel ved papirprototyper vil kompromisset være at enheten egentlig ikke fungerer. Ved høynivå prototyper kan kompromisser være at responstiden er større, utseende er dårligere eller at kun begrenset funksjonalitet er tilgjengelig.

To vanlige kompromisser som må balanseres er bredden av funksjonaliteten som gis kontra dybden. Vi skiller mellom to typer prototyping:

- **Horisontale prototyper** = viser totalsystemet uten særlig mye interaktivitet og funksjonalitet (dvs. mye funksjoner, men lite detaljer)
- **Vertikale prototyper** = går i dybden på en begrenset del av systemet og implementerer tilstrekkelig interaktivitet og funksjonalitet for å kunne teste denne (dvs. lite funksjoner, men mye detaljer)



Et annet kompromiss er mellom implementasjonstid og graden av funksjonalitet som representeres i prototypen. **Jo lengre implementasjonstid, desto mer funksjonalitet vil prototypen representere** (se figur). Høynivå prototyper krever altså større implementasjonstid enn lavnivå prototyper.

### Prototype kontra endelig produkt

Prototyper vil gå igjennom omfattende brukerevaluering, men forskjellen fra det endelige produktet er at produktet må gå igjennom omfattende kvalitetstesting for andre egenskaper slik som robusthet og feilfri operasjon. Det krever en annen testing for å bygge et produkt som kan brukes av tusenvis eller millioner av brukere, sammenlignet med en prototype som testes av noen brukere for å svare på spesifikke spørsmål.

### Lag-og-kast kontra evolusjonær utvikling

**Et alternativ til lag-og-kast prototyper er at man kan utvikle systemet evolusjonært, ved å la hver versjon bli testet og bygget videre på.** Fordelen med evolusjonær utvikling er at man kan bygge videre mot et ferdig produkt, mens ulempene er at koden fort blir veldig uryddig, det tar lengre tid å utvikle med tanke på at koden skal kunne utvides og det begrenser valgmuligheter i forhold til verktøy.

# F61 – Brukertestning



Læringsmål:

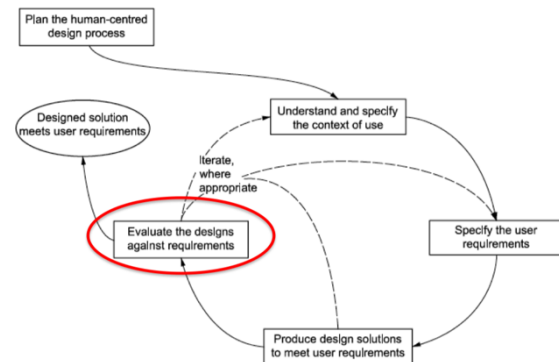
- **Forstå hensikten med brukertestning**
- **Bli kjent med brukbarhetslaboratoriet (UX-laboratoriet) mht. utforming og hva som inngår av utstyr**
- **Forstå stegene i brukertestning og deres hensikt**

## Brukertestning måler brukskvalitet <sup>(F)</sup>

Brukskvalitet er i hvilken grad produktet kan brukes av bestemte brukere for å oppnå bestemte mål i en bestemt brukskontekst med anvendbarhet, effektivitet og tilfredshet.

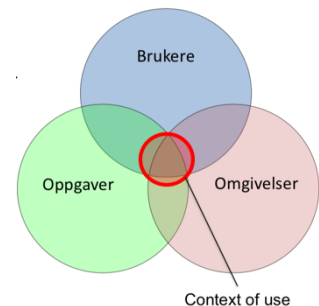
**En brukertest brukes for å måle brukskvaliteten, altså er det en brukerevaluering av produktet. Hensikten med en brukertest er å måle:**

- **Anvendbarhet** – i hvilken grad produktet lar brukeren utføre oppgavene. Brukertesten ser om systemet dekker de relevante funksjonene og om brukerne får til å bruke disse.
- **Effektivitet** – hvor effektivt oppgavene kan utføres med systemet. Brukertesten vil ta kvantitative mål, for eksempel hvor lang tid det tar å løse en oppgave
- **Subjektiv tilfredsstillelse** – den opplevde brukskvaliteten. Brukertesten finner brukerens opplevelse av systemet vha tester, intervjuer, feltstudier, spørreskjema, osv. Dette er viktig for brukeraksept.



For at det skal være meningsfullt å måle brukskvaliteten må vi kunne svare på hvem brukerne er, hva de ønsker å bruke produktet til (mål) og hvor produktet skal brukes. **Dette skyldes at brukskvaliteten er en relasjon mellom produktet og konteksten det brukes i, altså er brukskvaliteten kontekststøt.** For at resultatet til brukertesten skal være gyldig må brukskonteksten ved brukertestingen derfor være lik brukskonteksten i den virkelige situasjonen der produktet skal brukes. Validiteten til brukertesten forutsetter at:

1. **Deltakerne er representative brukere**
2. **Testoppgavene er representative i forhold til brukernes mål**
3. **De fysiske og sosiale omgivelsene der testen gjennomføres er representative i forhold til de reelle bruksomgivelsene**



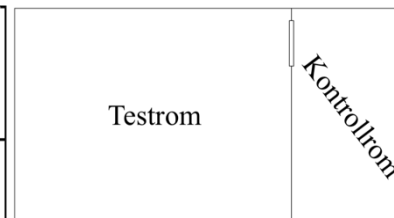
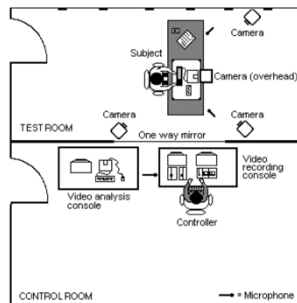
## Formativ og summative evaluering

Vi skiller mellom:

- **Formativ evaluering** – brukes i tidlige faser av designprosessen for å få innsikt i hvordan designet skal formes. Fokuset er å identifisere feil som brukerne oppdager, slik at dette kan formidles til utviklerne. Denne evalueringen kan føre til at store endringer skjer raskt.
- **Summativ evaluering** – brukes i sene faser av designprosessen for å godkjenne eller måle brukskvaliteten til et produkt (akseptansetest). Fokuset er målbare kriterier som for eksempel oppgavegjennomføring, feilrate, tidsbruk og tilfredsstillelse.

## UX-laboratoriet (F/B)

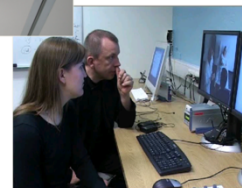
**Brukbarhetslaboratoriet (UX-lab) består av et testrom med opptaksutstyr og produktet som skal testes og et kontrollrom der evaluatorene kan se hva som foregår og analysere dataen.** Testrommet kan ordnes for å etterligne egenskaper ved den virkelige brukskonteksten, for eksempel kan det settes opp som et kontor. Det er lydtett, har ingen vindu og mobiltelefoner er ikke tillatt for å minimere forstyrrelser, slik at brukeren kan fokusere på de gitte oppgavene.



Kameraer og mikrofoner i taket.



Testrom



Kontrollrom



Styrbare kameraer i taket



Trådløse mikrofoner på brukerne



Trådløst mini-kamera for brille



Faste mikrofoner i rommet.



"Spelling" av PCer og mobile enheter



"Spelling" ved hjelp av KVM over IP

Som regel blir flere kameraer brukt for å filme brukerens oppførsel, slik som handbevegelser, ansiktsuttrykk og kroppsspråk. Mikrofoner brukes for å fange opp det brukeren sier. Video og annen data gis til monitorer i kontrollrommet, som ofte er separert fra testrommet vha et enveis-speil. Figurene viser andre typer utstyr som kan brukes. Mye av utstyret er trådløst og kan brukes i mobile brukbarhetslaboratorier, der utstyret kan settes opp midlertidig på et vanlig kontor.

Eye tracking



## Steg i brukertesting (F)

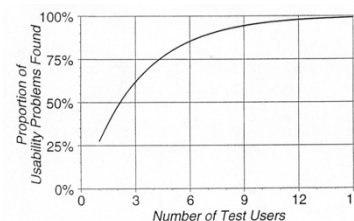
Følgende er stegene i brukertesting:

### 1. Utvikle målsettinger og hypoteser for testen, og utvikle testplan

Finn **hensikten med testen, hva resultatene skal brukes til, hvilke ressurser som trengs og når testen skal gjennomføres.** Det må også avgjøres **hvem som er ansvarlig.**

### 2. Skaff brukere gjennom tilfeldig eller stratifisert utvalg

**Bestem hvilke brukerskarakteristikker som er viktig,** for eksempel personlig historie (alder, kjønn, holdning), utdanningshistorie, erfaring med IKT, produkterfaring, jobbhistorie, osv. **Antall brukere** vil avhenge hvor objektiv testen må være, hvor mye ressurser man har til rådighet, tilgjengelighet av brukere og varigheten av testen. Som regel holder det med 5-8 brukere siden det er tilstrekkelig for å oppdage de fleste brukbarhetsproblemene (se figur).

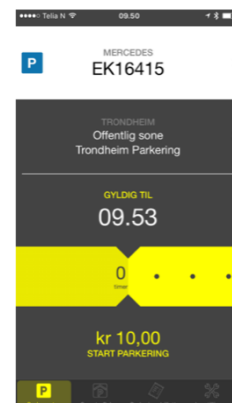


### 3. Forbered materiale og kontekst

**Det er viktig at testen forberedes godt** ved at man bestemmer hva som skal testes (papirprototype eller programvare), hvilke omgivelser som skal brukes (kontor, kafe, butikk, osv.) og hvilke oppgaver brukerne skal løse. **Denne beskrivelsen av situasjonen og oppgaver er et scenarie, og det skal lages så realistisk og komplett som mulig.** Viktige krav for scenariet:

- **Rekkefølgen på oppgavene** skal tilsvare det brukeren ville ha gjort i virkeligheten.
- Scenariet skal **matches opp mot brukeren** ved at noviser får enkle scenarier og eksperter får mer kompliserte
- Scenariet skal **gi oppgaver** som brukeren skal løse (eks: melde seg på et løp)
- Presenter oppgavene i en **målrettet form** med et **enkelt språk.**

- **Unngå å gi hint til oppgavens løsning ved å unngå bruke ord som tilsvarer navnet på funksjonen du vil de skal bruke** (eks: ikke bruk «start en parkering» for Smartpark appen)



For eksempel for Smartpark appen kan et scenarie være «Klokken er 11 en lørdag formiddag og du kjører en tur i midtbyen for å handle klær. Du har ikke tid til å bruke mer enn to timer på handlerunden. Du velger å parkere bilen i Munkegata. Frø du går ut av bilen kommer du på at du må løse inn parkeringsbillett. Hvordan vil du løse dette med appen». Denne inneholder en beskrivelse av omgivelsene og hvilke oppgaver som skal løses på et enkelt språk uten at det er gitt hint til hvordan oppgaven skal løses.

#### 4. Velg forsøksleder

Forsøkslederen kan være:

- **Intern UX-spesialist:** har akseptabel objektivitet og kunnskap om produktet
- **Markedsføringsspesialist/tekniker:** har god kjennskap til produktet, men lav objektivitet
- **Ekstern konsulent:** har god objektivitet og profesjonalitet, men lav kunnskap om produktet

#### 5. Pilottest

Det er viktig å gjennomføre en pilottest for å oppdage **svakheter ved metodikken og testdesignet**. Det gir også testteamet en **mulighet til å øve seg**.

#### 6. Utfør brukertesten

Utføring av brukertesten involverer 10 retningslinjer

Steg	Hensikt	Forklaring
1. Introduser deg selv	Skaper trygghet	Håndhils som vanlig og fortell hvem du og de andre i teamet er
2. Beskriv hensikten med testen	Skaper forståelse for våre forventninger og behov	Fortell at vi tester produktet og ikke dem, vi er interessert i brukervennligheten til produktet og ikke deres ferdigheter, vi trenger et friskt syn og hjelp til å teste produktet.
3. Fortell deltakerne at de kan avbryte når de vil	Skaper trygghet, tillit og følelse av kontroll	Informere brukeren om at testen er frivillig og de har full kontroll. Det finnes ingen skjulte motiver og de kan avbryte når som helst.
4. Beskriv utstyret i rommet og begrensningene til prototypen	Skape fokus på testen	Forklar hva utstyret brukes til og hva som ikke er en del av testen
5. Lær bort hvordan man «tenker høyt»	Få innsikt i brukerens mentale modell	Det er viktig at brukerne verbaliserer det de tenker, slik at vi får innblikk i de indre prosessene som ligger bak brukerens handlinger. Lær bort hvordan man tenker høyt.
6. Forklar at du ikke kan tilby hjelp under testen	Få brukerens mening	Forklar at du ikke kan svare på spørsmål underveis fordi vi ønsker brukerens mening. Det blir anledning til å spørre og diskutere etterpå.
7. Beskriv oppgaven og introduser produktet	Informere bruker om produkt og kontekst	Beskriv produktet og situasjonen. Ikke avslør funksjonalitet eller begrep som du ønsker tilbakemelding på. Gi brukeren en oppgaveliste.
8. Spør om det er noe de lurer på og kjør testen	Kjøre testen	La brukeren komme til orde før testen starter, og deretter kjør testen som observatør. Spørsmål reflekteres tilbake til bruker. Ikke grip inn når ting går galt, la brukeren få bruke tid. Hjelp brukeren til å tenke høyt og ta detaljerte notater (observasjonsskjema) eller film
9. Avslutt testen med å la brukeren uttale seg før du samler eventuelle løse tråder	Samle tråder	Når alle oppgaver er avsluttet eller tiden er oppbrukt skal testen avsluttes. Svar på spørsmål som ble stilt underveis, diskuter og be om subjektiv vurdering og eventuelle forslag til redesign.
10. Bruk resultatene	Finn input til redesign	Etter testen er gjennomført skal «breakdowns» (sammenbrudd) analyseres, problemene skal listes og prioriteres og man må finne årsakene til de viktigste problemene og lete etter alternative løsninger.

## 7. Omform data til funn og anbefalinger

Etter brukertesten er gjennomført må man analysere resultatet. **Feilhandlinger og problemer identifiseres ved å loggføre «breakdown»** i sanntid (direkte observasjoner) eller retrospekt (videoanalyser). Deretter må man prøve å **finne årsaken bak problemene**. Det er viktig å **prioritere funnene**, siden brukbarhetsproblemer kan være avhengig av hverandre. Videre kan man **utvikle nye løsningsforslag**, indikere hvor man trenger **ytterligere forskning** og **produsere en testrapport**.

### Viktige aspekter ved brukertesting

Noen viktige aspekter man må tenke på:

- **Bruk av spørreskjema** – brukeren kan bes om å fylle ut et spørreskjema etter testen. Det finnes flere standardiserte skjema, for eksempel **SUS** (System Usability Scale). SUS vil gi et tall mellom 0 og 100 som representerer opplevd brukervennlighet og det brukes i summativ evaluering. Det kan for eksempel være 10 påstander om systemet som brukeren har prøvd.
- **Når brukeren har problemer** – problemer skal ikke korrigeres der og da. Brukeren skal ikke få hjelp med mindre de er veldig forvirret, oppgaven gjør brukeren ukomfortabel, brukeren er så frustrert at hen nesten gir opp eller systemet har bugs, manglende kobling eller krasjer. Hvis man hjelper brukeren kan dette påvirke hva som skjer senere i forsøket. Det er viktig å ikke skyld på brukeren eller forklare hvordan systemet virker (dvs. den konseptuelle modellen)
- **Interaksjon med brukeren** – ikke vis at du er overrasket, fokuser på hva brukeren forventer når hen utfører en handling og hold deg i bakgrunnen. Aldri hint til at brukeren gjør feil eller jobber sakte. Ikke la brukeren få en følelse av dine preferanser. Vær oppmerksom på hvordan du sier ting og ditt eget kroppsspråk (det skal lite til før brukeren aner hva du mener).
- **Debriefing med bruker** – begynn med å la brukeren uttrykke seg. Diskusjonen bør starte på et høyt nivå før man beveger seg mot mer konkrete aspekter. Fokuser først på å forstå problemene istedenfor å løse dem. Husk å være upartisk under diskusjonen, slik at brukeren ikke føler at hen må forsvare handlingene sine.
- **Andre punkter:** gi lette oppgaver i starten, presenter en oppgave av gangen, hold en avslappet atmosfære med lite forstyrrelser, minimer antall observatører i rommet, osv.

**Noen spørsmål om systemet du har brukt.**

Vennligst sett kryss i kun en rute pr. spørsmål.

	Sterkt uenig							Sterkt enig
1. Jeg kunne tenke meg å bruke dette systemet ofte.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2. Jeg synes systemet var unødvendig komplisert.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Jeg synes systemet var lett å bruke.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Tolkning av sammenbrudd

Det er tre grunner til sammenbrudd («breakdowns»):

1. **Manglende funksjonalitet**
2. **Manglende kommunikasjon om funksjonalitet**
3. **Feil målgruppe** (kunnskapskrav, fysiologi, osv.)

For å tolke disse kan man observere bruken, intervju brukeren etterpå eller bruke egne erfaringer

### Etiske aspekter

**Brukbarhetstesting er ikke farlig for forsøkspersonene, men de blir utsatt for et psykisk press.** For å minimere dette er det viktig å ha alt klart når brukeren kommer, understrek at det er systemet som blir testet og ikke brukeren, informer om at de kan trekke seg når som helst, introduser utstyret, forklar at resultatet er konfidensielt, osv.

## 14.2 Brukstesting <sup>(B)</sup>

Målet ved en brukstest er å se om produktet som blir utviklet kan brukes av de bestemte brukerne for å utføre oppgavene som produktet har blitt designet for. En sentral komponent av brukertesten er å samle data om prestasjonen til brukerne når de gjennomfører forhåndsbestemte oppgaver. Det blir ofte brukt flere metoder for å samle data, for eksempel videoopptak av brukernes oppførsel, observasjon av bruker som tenker høyt, spørreskjema eller intervju etter interaksjon for å identifisere subjektive meninger, osv. Det blir ofte utført kvantitative mål, for eksempel hvor lang tid det tar å utføre en oppgave, antall feil, type feil, antall feil ved en bestemt oppgave, antall brukere som fullfører en oppgave suksessfullt, osv.

## 13.1-13.3 Evaluering <sup>(B)</sup>

Evaluering samler informasjon om brukernes opplevelse når de interagerer med systemet og formålet er å forbedre designet. Evalueringen fokuserer på både brukskvaliteten til systemet og brukeropplevelsen ved å interagere med det. Designere bruker evaluering for å sjekke at deres design aksepteres av et større publikum og ikke bare av de selv. Det finnes mange ulike typer evaluering og hvilken som brukes avhenger av målet ved evalueringen. Det vil som regel gå ut på å observere brukere og måle deres prestasjoner i brukertesting, eksperimenter eller feltstudier.

For å forstå evaluering må vi se på:

- **Hvorfor evaluere:** i dag forventer brukerne at systemet ikke bare er brukbart, men også fornøytelig å bruke. Produkter som er godt designet selger bedre, evaluering gjør at problemer kan fikses og designere slipper å diskutere hva de liker eller ikke liker
- **Hva skal evalueres:** vil variere fra lavnivå prototyper til fullstendige system, fra estetisk design til sikkerhetstrekk, osv.
- **Hvor skal det evalueres:** hvor evalueringen foregår vil avhenge av hva som evalueres. Laboratoriet brukes hvis man trenger mye kontroll, mens feltstudier brukes hvis man trenger realistiske settinger. Indirekte observasjoner kan brukes for å observere interaksjon hos flere deltakere i deres naturlige omgivelser uten forstyrrelser.

Tre typer evalueringer:

1. **Kontrollerte settinger med brukere** – aktiviteten til brukerne kontrolleres for å teste hypoteser eller måle bestemte oppførsler. Den vanligste er brukstesting. Kontrollerte settinger gjør at evaluatorene kan kontrollere hva brukerne gjør, når de gjør det og hvor lenge de gjør det. Forstyrrelser og utenforstående påvirkninger kan reduseres.
2. **Naturlige settinger med brukere** – det er lite eller ingen kontroll over aktiviteten til brukerne for å bestemme hvordan produktet brukes i den virkelige verden. Den vanligste er feltstudier. Målet er å være diskre, slik at man ikke påvirker hva brukerne gjør i løpet av evalueringen.
3. **Enhver setting uten brukere** – konsulenter og forskere undersøker aspekter ved grensesnittet for å identifisere brukbarhetsproblemer. For eksempel kan det være inspeksjoner, heuristikk, modeller og analyse. Evaluatorene må se for seg hvordan grensesnittet sannsynligvis vil brukes basert på kunnskap om brukskvalitet, brukernes oppførsel, brukskonteksten og type aktivitet som skal utføres.

Det er fordeler og ulemper ved hver type. For eksempel vil lab-baserte studier være gode til å avsløre brukbarhetsproblemer, men dårlige til å fange brukskonteksten. Feltstudier er gode til å demonstrere hvordan mennesker bruker teknologien i den tiltenkte settingen, men de er dyre og vanskelige å utføre. Undersøking av aspekter uten brukere er billigere og lettere, men kan føre til at noen brukbarhetsproblemer ikke oppdages. Ofte vil kombinasjoner av metodene brukes.<sup>61</sup>

## ISO 9241-210 – Brukersentrert evaluering

Det er flere ulike brukersentrerte metoder som kan brukes for å evaluere design. For å oppnå gyldig resultat, bør evalueringen utføres av erfarne evaluatorene og passende metoder bør brukes. Brukersentrert evaluering kan brukes ved alle stegene i prosjektutviklingen, men jo senere i utviklingen, desto dyrere blir endringene. **I tidlig stadier brukes formativ evaluering for å få feedback som kan brukes for å forbedre produktet, mens i senere stadier blir summativ evaluering brukt for å sjekke at brukerkravene har blitt tilfredsstillende.** To brukersentrerte evalueringer er brukertesting og inspeksjonsbasert evaluering.

### Brukertesting

Brukertesting kan brukes i alle stadier av designprosessen. **Ved formativ evaluering tidlig i designprosessen kan brukere gis modeller, scenarier eller skisser av designkonsept og bes om å evaluere disse mot en reell kontekst.** Denne evalueringen kan gi verdifull feedback på det foreslåtte designet. Papirprototyper kan brukes for å evaluere detaljerte aspekter ved designet på en rask og billig måte. Når brukerne tester prototyper bør de utføre oppgaver med prototypen og informasjonen som samles brukes for å drive designprosessen. **Ved summativ evaluering senere i designprosessen kan brukertesting benyttes for å se om målene for brukskvalitet (anvendbarhet, effektivitet og tilfredshet) har blitt oppnådd i den gitte brukskonteksten.** Brukertester kan utføres på lab (kontroll) eller i feltet (realisme).

### Inspeksjonsbasert evaluering med retningslinjer for brukskvalitet

Inspeksjonsbasert evaluering kan være verdifull og kostnadseffektiv, og det kan komplementere brukertesting. **Det kan brukes for å eliminere hovedproblemer før brukertesting, slik at brukertesten blir mer kostnadseffektiv.** Denne evalueringen blir utført av eksperter på brukskvalitet som baserer vurderingen sin på tidligere erfaring med brukbarhetsproblemer og diverse retningslinjer og standarder (eks: designprinsipper og heuristikk). Ofte vil flere eksperter vurdere produktet, slik at man unngår individuell bias. Effektiviteten til inspeksjonen vil avhenge av egenskapene, erfaringen og kunnskapen til evaluatorene.

Inspeksjonsbasert evaluering er enklere og raskere å utføre, og det kan ta hensyn til et bredere aspekt av brukere og oppgaver enn brukertesting. Inspeksjonen vil likevel sjeldent finne alle brukbarhetsproblemene, og bør derfor kombineres med andre evalueringsmetoder.







# F70 – Mobil IKT: Brukskvalitet og brukstesting

Læringsmål:

- **Brukbarhetsutfordringer knyttet til mobil IKT**
- **Mobile interaksjonsteknikker**
- **Krav til mobil IKT setter til brukertesting (vis-a-vis brukertesting av stasjonære løsninger)**

## Mobil IKT – Ubiquitous computing <sup>(F)</sup>

**Den tredje bølgen i interaksjonsparadigmene er mobil og ubiquitous (allestedsværende) computing.** IKT har gått fra å være en stasjonær datamaskin med én bruker i en relativt forutsigbar brukskontekst, til å bli en mobil teknologi som brukes i svært mange ulike situasjoner. Det er mange typer datamaskiner og sensorer på nett, og de er integrert i rom, klær og hverdagslige ting. De støtter derfor menneske-maskin interaksjon i ulike brukskontekster.

	Forholdet bruker–datamaskin	
Mainframe computing (First wave, ca. 1960-1985)		
Personal computing (Second wave, ca. 1985-2000)		
Mobile/Ubiquitous computing (Third wave, ca. 2000-present)		

**Visjonen til ubiquitous computing (UbiComp) er at interaksjon med datamaskiner skal foregå hvor som helst og når som helst.** Teknologien er alltid tilstede i brukerens bevissthet, fordi den er sømløst integrert i våre liv og fungerer som en transparent «forlengelse» av kroppen vår. **For at denne visjonen skal realiseres må løsningene tilpasses mobil bruk, altså brukerne må bli mobile, utstyret må bli mobilt og løsningene må kunne brukes i ulike fysiske og sosiale omgivelser.** I tillegg er det krav på tilstrekkelig kvalitet på sensorer og maskinvare siden dette påvirker brukeropplevelsen ved interaksjonen. Det må også etableres digitale økosystem som tilbyr kompatibelt utstyr og ulike applikasjoner.

## Mobilt er ikke nok <sup>(F)</sup>

**For at datamaskiner skal bli allestedsværende er det ikke tilstrekkelig at løsningene er mobile.** For eksempel ble det utviklet flere touch-mobiler før det virkelig ble populært blant brukerne. **De mobile løsningene må også ha tilstrekkelig brukskvalitet, noe som forutsetter:**

- **Tilpassede konseptuelle modeller**
- **Tilpassede brukergrensesnitt**
- **Tilpassede interaksjonsteknikker**



**Mobile designløsninger må tilpasses mobilt bruk for å sikre høy brukskvalitet.** Dette krever at designere har en god forståelse av den fysiske og sosiale brukskonteksten. Den konseptuelle modellen og brukergrensesnittet må tilpasses den mobile bruken og det må benyttes egnede interaksjonsteknikker. **Kravet for å oppnå ubiquitous computing er altså at de mobile løsningene har høy brukskvalitet og gir gode brukeropplevelser.** For å oppnå dette må designløsningene tilpasses mobilt bruk.

## Interaksjonsteknikk <sup>(F)</sup>

**Interaksjonsteknikk er en metode som gjør at brukeren kan gjennomføre relevante oppgaver på en datamaskin.** Det vil ofte innebære en kombinasjon av input og output, og



tilhørende maskinvare og programvare. Et eksempel er Drag-and-drop der brukeren i et vindussystem kan flytte elementer ved å trykke og dra dem vha musen. Elementene vil flyttes når brukeren flytter på musen. **Interaksjonsteknikker kan ha stor betydning på brukskvaliteten til en løsning**

Interaksjonsteknikk	Egnet for:	Uegnet for:
PC, mus og tastatur	<ul style="list-style-type: none"> <li>Kontoromgivelser</li> <li>Dokumentbehandling</li> <li>Situasjoner hvor store skjermflater er viktig.</li> </ul>	<ul style="list-style-type: none"> <li>Mobile brukssituasjoner</li> <li>Mange samtidige brukere</li> <li>Brukere som ikke har frie hender.</li> </ul>
Touch (berøring)	<ul style="list-style-type: none"> <li>Mobile brukssituasjoner</li> <li>Brukere som trenger å ha hendene frie (f.eks. bilførere, kirurger, etc.)</li> <li>Kan støtte opp om flere samtidige brukere</li> </ul>	<ul style="list-style-type: none"> <li>Presisjonsarbeid</li> </ul>
Pennbasert-interaksjon	<ul style="list-style-type: none"> <li>Mobile brukssituasjoner som krever håndpresisjon (f.eks. frihåndstegning, skrive for hånd)</li> </ul>	<ul style="list-style-type: none"> <li>Brukere som ikke har frie hender</li> </ul>
Stemmestyring (f.eks. "Siri")	<ul style="list-style-type: none"> <li>Brukere som trenger å ha hendene frie (f.eks. bilførere, kirurger, etc.)</li> <li>Stille omgivelser</li> </ul>	<ul style="list-style-type: none"> <li>Komplekse oppgaver (må huske kommandoer)</li> <li>Mange samtidige brukere</li> </ul>
Gest-basert (f.eks, peking på Kinect)	<ul style="list-style-type: none"> <li>Mange samtidige brukere</li> <li>Brukere som trenger frie hender</li> </ul>	<ul style="list-style-type: none"> <li>Presisjonsarbeid</li> </ul>

Styrker og svakheter ved ulike interaksjonsteknikker

	Stasjonær (PC)	Mobil (smart-telefon)
Forgrunn / Bakgrunn	Hovedfokuset er på skjermen.	Mobiltelefonen er integrert med andre aktiviteter.
En bruker/ Flere brukere	Oftest én bruker. Samhandling over nett.	Flere brukere samtidig, i den fysiske verden.
Omgivelsesdata	Fysiske posisjon er irrelevant (CyberSpace)	Lokasjon og sensordata kan brukes av software.
Formfaktor/ergonomi	PCen har funnet sin fysiske form (desktop/laptop).	Formfaktor blir viktig (størrelse, vekt, form, materialer, batterilevetid, robusthet)
Interaksjon	Store skjermer, tastatur og mus	Små skjermer, penn, knapper, berøring, stemmestyring, kroppen som interaksjonsmedium

Stasjonær vs. mobil interaksjon

Som vi kan se av tabellen til venstre vil det være styrker og svakheter ved de ulike interaksjonsteknikkene. For eksempel vil touch- og gest-basert interaksjon være uegnet for presisjonsarbeid, men egnet for brukere som trenger frie hender eller samtidig bruk. Hvilke interaksjonsteknikker designløsningen legger til rette for, kan ha stor påvirkning på hvordan brukerne opplever å interagere med produktet.

## Brukertestning av mobil IKT (F)

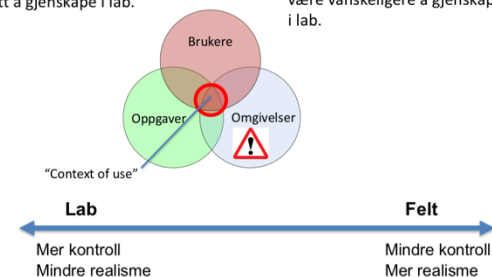
Skrivebordssituasjonen er relativt lett å gjenskape i lab, mens mobile brukssituasjoner kan være vanskeligere å gjenskape. **Brukertestning av mobil IKT setter høye krav til testomgivelsene, fordi det kan være utfordrende å gjenskape mobile brukssituasjoner i lab og det kan være vanskeligere å balansere behovene for kontroll og realisme. Det er mulig å simulere mobile brukskontekster i lab for å gi mer realisme, men dette vil føre til mindre kontroll.**



«Skrivebordssituasjonen» er relativt lett å gjenskape i lab.



Mobile brukssituasjoner kan være vanskeligere å gjenskape i lab.



## Viktige faktorer for opplevd brukskvalitet

Viktige faktorer for opplevd brukskvalitet:

- **Løsningen må ikke kreve for mye av brukerens oppmerksomhet**
- **Løsningen må ha en forutsigbar oppførsel (eks: synlighet og presisjon)**
- **Løsningen må kunne integreres med andre elementer i hverdagen/arbeidet**
- **Løsningen må ha passende fysiske og sosiale aspekter i brukssituasjonen**
- **Løsningen må tilby riktige interaksjonsteknikker (eks: handsfree, touch, osv.)**

# F71 – Deltagende design



Læringsmål:

- Danne en overordnet forståelse av hva deltagende design er
- Forstå den sosiale, politiske og etiske dimensjonen av teknologi:
  - Verdier i teknologi
  - Brukermedvirkning
  - *Tool perspective*
- Forstå forskjeller på brukersentrert og deltagende design

Kvalitetskriterier ved design <sup>(F)</sup>

Tre kvalitetskriterier ved design er:

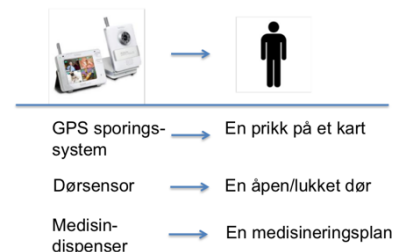
- **Utilitas** – at det fyller et behov
- **Firmitas** – at det er teknisk godt
- **Venustas** – at det er vakkert



Den etiske, politiske og sosiale dimensjonen av IKT <sup>(F)</sup>

**Teknologi vil påvirke hvordan vi «ser» verden.** For eksempel i velferdsteknologi kan GPS sporingssystem gjøre at pasienten ses på som en prikk på et kart istedenfor et menneske.

**Dette illustrerer at datamaskiner kan føre til en innsnevring av fokuset.** Den etiske, politiske og sosiale dimensjonen av IKT er viktig innenfor tema som AI og autonome våpen og biler, myndighetenes aksess til mobiltelefonlogger, elektronisk logging av helsedata, Piratebay vs. Apple iTunes Store, osv.



Det er ulike teorier på hvordan verdier bygges inn i teknologi:

- **Teknologien representerer designerens verdisyn**
- **Teknologien representerer samfunnets verdisyn**
- **Verdier i teknologi og i samfunn interagerer (påvirker hverandre)**

Deltagende design <sup>(F)</sup>

**Deltagende design tar utgangspunkt i den politiske, sosiale og etiske dimensjonen av teknologi, og er basert på at brukere har rett til å delta aktivt i beslutninger som påvirker deres liv.**

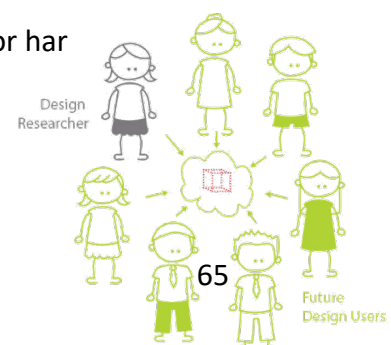
**Brukermedvirkning (user empowerment) er sentralt for at teknologi skal støtte opp demokratisering og maktfordeling.** Deltagende design har altså fokus på at brukerrepresentanter skal være en del av designteamet, slik at de får en større påvirkning på designprosessen.

**Deltagende design er basert på brukermedvirkning, demokratisering og maktfordeling.**

Deltagende vs. brukersentrert design

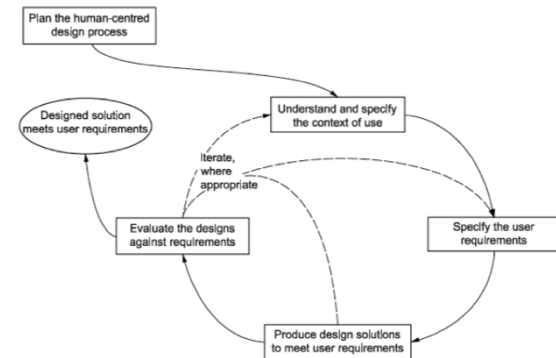
To ulike designmetoder:

- **Deltagende design** = opprinnelse i Skandinavia på 1970-tallet. **Det er politisk motivert og basert på brukermedvirkning.** For eksempel har økt automatisering skapt frykt for tap av industriarbeidsplasser, så derfor har industriarbeidere fått medvirke i design av arbeidsverktøy for å kunne redde arbeidsplasser. **Designeren opptrer som fasilitator og brukerrepresentanter er en del av designteamet.** Løsningen



designes i stor grad av brukerne gjennom deltagende designaktiviteter (eks: workshops). **Det er fokus på brukermedvirkning** og brukskvalitet er sekundært.

- **Brukersentrert design** = opprinnelse i den amerikanske forretningsverden på 1980-tallet (Xerox Parc). **Det er basert på systematisk gjennomføring av brukertester for å lage mer brukervennlige produkter som gir større inntjening. Designeren er i større grad beslutningstaker og det er et tydelig skille mellom designer og bruker.** Løsningen designes for brukere basert på innsikt fra for eksempel brukertester. **Det er fokus på brukskvalitet.**



**Mange av teknikkene vi bruker i brukersentrert design har sitt opphav i deltagende design.**

Brukersentrert design utføres på vegne av brukerne, mens deltagende design utføres med brukerne.

### Tool perspective

**Tool perspective påstår at teknologi bør ta utgangspunkt i eksisterende praksis**, og at nye verktøy bør komplementere og videreutvikle eksisterende praksis istedenfor å erstatte den.

**Man skal la brukere dra nytta av og videreutvikle taus kunnskap, som er erfaringsbasert.**

Taus kunnskap er lært ved bruke intuisjon (erfaring) og det er vanskelig å formalisere og automatisere. For eksempel kan mekanikeren høre på lydene at noe ikke stemmer.

### Deltagende design – verktøy og teknikker

**I deltagende design blir det brukt verktøy og teknikker som er lette å forstå for alle deltakere og som muliggjør rask prototyping som legger til rette for engasjement og refleksjon.** Eksempler er co-design av lavnivå prototyper, rollespill, service blueprints, personas, storyboard, osv.

### Boks 11.5 – Involvering av brukere i design <sup>(B)</sup>

Ideen om deltagende design har opprinnelse i Skandinavia i slutten av 1960-tallet og starten av 1970-tallet. Det var to drivkrefter: ønsket om å kommunisere informasjon om komplekse system og unionsbevegelsen som presset på at arbeiderne skulle ha demokratisk kontroll over endringer i deres arbeidsliv. Flere prosjekter forsøkte å involvere brukerne i designet og fokuserte på arbeidet istedenfor å produsere et produkt. Det er ikke lett å involvere brukerne i designvalg pga. kulturelle forskjeller (eks: brukere klarer ikke å koble kunnskap og erfaring til det de blir fortalt). Mock-ups og simuleringer av arbeidssituasjonen blir brukt for å gjøre det lettere for arbeiderne å bruke deres erfaringer og taus kunnskap og for designerne å forstå de faktiske oppgavene arbeiderne må utføre.

### Metoden for deltagende design <sup>(Spinuzzi)</sup>

Deltagende design er basert på metoder som sikrer at deltakernes tolkninger blir tatt hensyn til i designet. **Deltakernes tolkning blir ikke bare brukt for bekreftelse, men som en essensiell del av designprosessen.** De tause, usynlige aspektene av menneskeaktiviteter blir produktivt og etisk undersøkt gjennom designpartnerskap med deltakere. **Designere og deltakere samarbeider med å designe artefakter, arbeidsflyt og arbeidsmiljø.** Dette samarbeidet blir gjennomført iterativt, slik at designernes og deltakernes forståelse av aktiviteten kan utvikles og forbedres. **Deltagende design forsøker å kombinere deltakernes**

**tause kunnskaper med designernes mer abstrakte, analytiske kunnskap.** Av politiske og etiske årsaker må disse to kunnskapene kobles sammen og verdsettes av alle som er involvert i designprosessen.

Deltagende design har fokus på demokrati og maktfordeling. Det ble utviklet på en tid der datateknologi ble introdusert til arbeidsplassen og tok over jobben til arbeiderne. Siden de ikke hadde kunnskap om hvordan datateknologiene kunne designes, ble arbeiderne tvunget til å akseptere disse teknologiene eller avslå dem. Deltagende design ble utviklet for å være et tredje alternativ der programvareutviklere og arbeidere kan samarbeide for å utvikle og forbedre ny teknologi, slik at arbeiderne får bevare kontrollen over deres arbeid.

### Taus kunnskap

Taus kunnskap blir utviklet og brukt av de som arbeider med teknologier. Det er vanskelig å formalisere og beskrive, og har ofte blitt ignorert i menneske-maskin interaksjon. Det politiske aspektet ved taus kunnskap er at arbeiderne bør ha et ord i deres arbeid fordi de vet best hvordan det skal utføres (demokrati på arbeidsplassen). Det er ikke mulig å fullstendig formalisere og klassifisere kunnskap, fordi den er tolkbar. Taus kunnskap er implisitt og ikke-systematiser. **Det er kunnskapen mennesker har som de ikke kan sette ord på.** Taus kunnskap er viktig i deltagende design, så derfor ble *Tool perspective* utviklet. Dette er en ide om at datateknologi blir designet som en samling av verktøy som den erfarne arbeideren kan bruke. Utviklingen av teknologi skal ta utgangspunkt i eksisterende praksis istedenfor detaljerte analyser, beskrivelser og formaliseringer av kvalifikasjoner.

**Et mål ved deltagende design er å bevare taus kunnskap, slik at teknologier kan passe inn i det eksisterende nettverket av taus kunnskap, arbeidsflyt og arbeidsverktøy, istedenfor å erstatte dem.** Deltakende design antar at taus kunnskap ikke kan fullstendig formaliseres, systematiseres eller kvantifiseres. Det blir ofte oversett og undervurdert, men kan være avgjørende for at aktiviteten skal bli suksessfull.

### Stegene i deltagende design

Deltagende design kan utføres på ulike måter, men tre grunnleggende steg er alltid tilstede:

1. **Initial utforskning av arbeidet** – designerne møter brukerne og blir kjent med hvordan brukerne arbeider sammen. Utforskningen inkluderer teknologiene som brukes, arbeidsflyten og arbeidsprosedyrer.
2. **Oppdagelsesprosess** – designere og brukere benytter ulike teknikker for å forstå og prioritere arbeidsorganisasjonen og den tiltenkte fremtidige arbeidsplassen. Brukernes mål og verdier blir spesifisert og man blir enig om ønsket utfall for prosjektet
3. **Prototyping** – designere og brukere vil iterativt forme teknologiske artefakter som passer inn i arbeidsplassen beskrevet i steg 2.

Disse stegene blir iterert flere ganger.

### Begrensninger ved deltagende design

Begrensninger ved deltagende design er:

- **Tillater ikke radikale endringer** som noen ganger må karakterisere nye system. Designet vil utvikle, men ikke revolusjonære systemet
- **Det tar mye tid, ressurser og institusjonelt forpliktelse**
- **Det gir lite struktur og krever at arbeidere kontinuerlig deltar**

## Evaluering av deltakende design

Deltagende design skal:

- **Forbedre kvaliteten til arbeidernes liv** = oppnås vha demokratisk brukermedvirkning (de får kontroll over deres arbeidsliv) og funksjonell medvirkning (deres evne til å utføre oppgaver blir bedre). Designerne og brukerne må interagerer vha intervju, fokusgrupper, workshops, osv. for å synkronisere deres tolkninger. **Designerne og brukerne må sammen bestemme kravene til prosjektet, slik at designere ikke tar total eierskap til prosjektet. Brukere får forme prosjektet til å reflektere deres verdier og mål.**
- **Samarbeid i utviklingen** = viktig for å utforske den tause kunnskapen og usynlige praksiser som ellers kan bli oversett. Det oppfordrer arbeiderne til å delta i deres egen brukermedvirkning. Det må brukes rutiner som sikrer at brukernes bekymringer blir adressert i det resulterende designet. Det er ikke mulig å inkludere alle arbeiderne, så representative arbeidere må velges.
- **Iterativ prosess** = taus kunnskap og usynlige praksiser er vanskelig å oppdage, så en iterativ prosess må følges for at det skal bli lettere for arbeidere å gi kritisk og effektiv respons på en prototype.



# F80 – Brukersentrert design i praksis

Læringsmål:

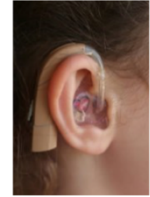
- Bygge en praktisk forståelse av den brukersentrerte designprosessen gjennom å følge en konkret design-case
- Gi innsikt i rasjonale tanker bak ulike metodevalg og designvalg
- Bedre forstå fasilitatorrollen

Design-case – problemstilling (F)

Vi skal se på brukersentrert design av interaktiv løsning for brukermedvirkning i høreapparattilpasning. Tap av hørsel kan ha alvorlige konsekvenser for det sosiale livet og arbeidslivet til en person. Til tross for dette er bruk av høreapparat relativt lite utbredt. Et høreapparat er et hjelpemiddel for personer med nedsatt hørsel og det er en liten lydforsterker som festes i øret. Et



Hørehorn



Analogt høreapparat



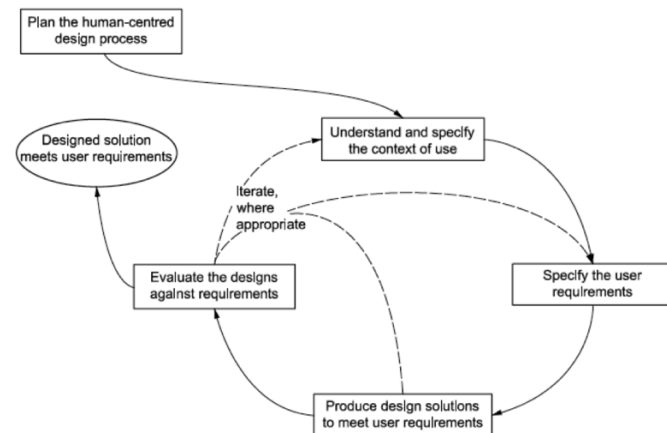
Digitalt høreapparat (m/smart-funksjonalitet)

sentralt mål i konsultasjoner hos audiograf er å øke tilfredsheten og fordeler ved bruk av høreapparat. I tilpasningsprosessen er det viktig med aktiv pasientdeltakelse, men det er også utfordrende å oppnå. Vi skal se hvordan interaktiv teknologi kan bidra til aktiv pasientinvolvering i høreapparattilpasning.



Brukersentrert designprosess (F)

Vi må begynne med å planlegge den brukersentrerte designprosessen. Denne prosessen krever et flerfaglig design- og utviklingsteam som består av to interaksjonsdesignere, en programmerer, en med kompetanse i audiologi og høreapparat og en prosjektleder. Det er også behov for tett samarbeid med brukerne som er pasienter med hørselstap og audiografer.



Første iterasjon

Den første iterasjonen har som mål å få innsikt i situasjonen og designe og evaluere en ikke-funksjonell papirprototype.

1. **Forstå brukskonteksten** – for å kunne lage en god løsning må vi først forstå hvorfor pasientinvolvering er utfordrende. For å få bedre innsikt bruker vi feltobservasjoner (hva skjer på en hørselskonsultasjon), intervju etter konsultasjon og litteraturstudie. De identifiserte barrierene for aktiv pasientdeltakelse var feil lytteomgivelse (stille kontor), det er vanskelig å beskrive lytteopplevelser og hørselsproblemer og det anvendte dataverktøyet er utelukket designet for audiologen.



2. **Spesifisering av brukerkrav** – for å identifisere brukerkravene ble det utført workshop med audiologer og pasienter. Resulterende krav var at ny løsning ikke bør gjøre at konsultasjonen tar lengre tid (45-60 min per pasient), det må redusere antall pasienter som må komme tilbake flere ganger for å få en god nok tilpasning, det bør kunne fungere med de fleste pasientene ved klinikken og løsningen må oppleves som



Rikere lydombegivelser



Delt brukergrensesnitt

verdifull for både audiograf og pasient. Den nye løsningen bør kunne støtte delt kontroll mellom pasient og audiolog, som kan oppnås vha et delt brukergrensesnitt. Løsningen bør også ha mer representative lydombegivelser.

**3/4. Design og evaluering:** co-design workshops brukes for å gjennomføre prototyping og evaluering samtidig. Det ble gjennomført 3 co-design workshops med varighet på 2,5-3 timer per workshop. Deltagerne var 2 audiografer, 2 høreapparatbrukere og 2 interaksjonsdesignere (moderatorer). Workshopen starter med en introduksjon og oppvarming, før det settes i gang gruppearbeid. Hver gruppe lager en lavnivå papirprototype og utspilling av et scenarie. Deretter følger presentasjon og diskusjon av løsninger.



Datainnsamlingen består av videopptak og foto av skisserte løsningsforslag. Analysen av dette gjorde at vi lærte at det er viktig å kunne **kontekstualisere hørselsproblemet, altså knytte det til en konkret lydombegivelse**. Dette kan gjøres vha en visuell og auditiv representasjon av dagligdagse lytteombegivelser (se figurer). Denne representasjonen kan fungere som en felles referanse for både audiograf og høreapparatbrukere, slik at man blir mindre avhengig av høreapparatbrukerens evne til å beskrive hørselsproblemet.

Lytteombegivelse: Ankomsthall



Lyttekilde

Støykilde

Braker legger en støykilde til i den simulerte lydombegivelsen



"Tap-to-play"

Det er også viktig med **individuell tilpasning**, siden ulike høreapparatbrukere kan ha ulike utfordringer i samme lytteombegivelse. Tilpasning av lytteombegivelser kan gjøre det lettere å imøtekomme individuelle behov. Involveringen av pasienten vil også avhenge av **formfaktor og ergonomisk utforming til løsningen**, for eksempel brukernes plassering rundt bordet, interaksjonen mellom brukerne og interaksjonen med bordløsningen.



Ansikt-til-ansikt

+ Muliggjør lesing på leppene.  
+ Øyekontakt, lett å ha dialog.

- Vanskeligere å se hva motparten gjør.  
- Mindre inkluderende (gjenspeiler konsultasjonssituasjonen)



Side-om-side

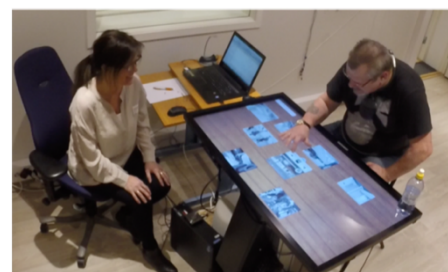
- Vanskeligere å lese på leppene.  
- Mindre øyekontakt; vanskeligere å ha dialog.

+ Lett å se hva motparten gjør.  
+ Mer inkluderende (gjør partene "likeverdige")

### Andre iterasjon

Den andre iterasjonen har som mål å lage konseptuell modell og design av en funksjonell prototype. Det skal også utføres en **evaluering av opplevd nytteverdi og brukskvalitet**. Figuren til høyre viser en mulig konseptuell modell som blir utviklet ved å se på brukerkonteksten og brukerkravene.

- 3. Produksjon av designløsning** – det blir laget høynivå funksjonelle prototyper i form av interaktive arbeidsflater som gir tilgang til lydopptak fra ulike lytteombegivelser
- 4. Evaluering av designløsning** – observasjoner og intervju (subjektiv nytte) blir brukt for å finne ut hvordan simulatoren påvirker interaksjonen mellom pasienten og audiologen under høreapparattilpasningen.



Interaktiv arbeidsflate med tilgang til lydopptak fra ulike lytteombegivelser.

12 pasienter og 3 audiologer deltok i testprosedyren og datainnsamlingen. Av utstyr ble det brukt en lyd simulator (interaktiv arbeidsflate), bærbar PC med pasientjournaler og programvare for programmering av høreapparat og Bluetooth hardware modul for trådløs programmering av høreapparat. Prosedyren besto av briefing før konsultasjon, selve konsultasjonen og konkluderende intervju. Datainnsamlingen består av video.

## Resultat

Resultatet var en iterativ, pasientdrevet tilpasningsprosess. Designløsningen komplementerer medisinsk informasjon og tiltak, og det gir økt pasientkontroll over tilpasningsprosessen. Bruk av simulatoren bidro til en mer iterativ tilpasningsprosess enn det som ble observert på klinikkene, mange feedback loops mellom pasient og audiolog og gradvis forfining av tilpasningen. Designløsningen la til rette for rikere kontekstspesifikke tilbakemeldinger fra pasienten, for eksempel «Lyden av bilen er forferdelige skarp». Det ble også lettere for audiologen å gi mer kontekstspesifikke oppfølgings spørsmål, for eksempel «Opplevs lyden fra bilen bedre nå?».

Simulatoren gjorde at pasienten kunne oppleve hva hørselsproblemet besto i og resultatet av tiltak. Den bidro til at pasienten fikk økt kontroll over hvilke lytteomgivelser høreapparatet skulle optimaliseres for og tempoet i tilpasningsprosessen. Den opplevde nytteverdien til pasienter er at det er verdifullt med en indikasjon på hvordan høreapparatet vil fungere i hverdagen, det er lettere å beskrive problemet og det vil potensielt redusere tidsforbruket i forbindelse med tilpasningen. Den opplevde nytteverdien for audiologer er at det er verdifullt med realistiske lydreferanser og det gir mulighet for en mer utforskende tilpasningsprosess.

Prototypen bidro til økt pasientinvolvering fordi lydopptakene fungerte som felles referanse i dialogen, som et verktøy for koordinering og formidling og som en allsidig løsning som kunne brukes i ulike sammenhenger. Prototypen fylte flere ulike funksjoner i konsultasjonen, for eksempel fungerte det som et referanseverktøy i forbindelse med tilpasningen, et verifikasjonsverktøy for diagnostiseringen, et forklaringsverktøy med pedagogisk verdi, osv.

## Identifisere brukbarhetsproblemer

Brukbarhetsutfordringer kan være sen respons ved trykking på skjerm, brukeraksept av simuleringene som følge av manglende realisme på noen av simuleringene og manglende retningsstyring på lyd (lyttekilder/støykilder).

## Konklusjon

Lyd simulatoren bidro til økt pasientinvolvering i forbindelse med tilpasning av høreapparat på flere måter. Responsen fra deltagende pasienter og audiologer var utelukkende positivt og det ble identifisert flere fordeler ved bruk. Hyppige endringer i behov understreker nødvendigheten av fleksibilitet i løsninger.