

# Midtsemesterrapport

## Metoder for evaluering av mobile brukergrensesnitt

Jo Christian Magnussen, Marius Oppedal, Daniel Lagutin, Øyvind Reistad



# User interfaces



## Innholdsfortegnelse

1. Innledning .....	4
2. Prosjektet .....	4
2.1 Hva skal vi gjøre? .....	4
2.2 Hvordan skal vi gjøre det? .....	4
2.3 Hva venter vi å finne? .....	5
3. Metodene .....	5
3.1 Brukbarhetstesting .....	5
3.1.1 Brukbarhet .....	5
3.1.2 Komponenter i brukbarhetstesting .....	6
3.1.3 Fordeler og ulemper .....	6
3.2 GOMS .....	6
3.2.1 Keystroke Level Model .....	7
3.3 Heuristisk Evaluering .....	8
3.3.1 Evalueringsprosessen .....	8
3.3.2 Fordeler med bruk av heuristisk evaluering .....	8
3.3.3 Nielsens heuristikker .....	8
4. Arbeidet hittil .....	10
5. Veien videre .....	10
6. Referanser .....	11
6.1 Brukbarhetstesting .....	<b>Feil! Bokmerke er ikke definert.</b>
6.2 GOMS: KLM .....	<b>Feil! Bokmerke er ikke definert.</b>
6.3 Heuristisk Evaluering .....	<b>Feil! Bokmerke er ikke definert.</b>
7. Vedlegg .....	12
7.1 Artikkel: "When two methods are better than one: combining user study with cognitive modeling" .....	12
7.2 Testplan: Brukbarhetstest .....	12

## 1. Innledning

Dette dokumentet, midtsemesterrapport i faget INF5261 Utvikling av mobile informasjonssystemer, er ment å skulle definere prosjektoppgaven vi jobber med. I tillegg vil vi forsøke å gi en innføring i evalueringsmetodene vi jobber med, si litt om hvor langt vi har kommet i prosjektet og hva vi kommer til å gjøre fremover.

Betydningen av et velfungerende brukergrensesnitt blir viktigere og viktigere. Spesielt gjelder dette for mobile enheter ettersom disse er usatt for en stadig konvergens, der flere og flere ulike teknologier veves inn i en og samme enhet. Det ligger en stor utfordring i å gjøre teknologiene i mobile enheter tilgjengelige og nyttige på brukernes premisser.

Det finnes metoder for en evaluering av brukergrensesnitt på stasjonære enheter, men det hersker fremdeles usikkerhet om disse metodene er gyldige for mobile enheter. Gjennom dette prosjektet håper gruppen å avdekke eventuelle svakheter og feil ved disse metodene, og gjennom dette finne ut om anvendelsen av disse på mobile enheter er hensiktsmessig.

## 2. Prosjektet

### 2.1 Hva skal vi gjøre?

Prosjektoppgaven er gitt av masterstudent Trenton Schulz. Oppgaven går ut på å evaluere mobiltelefoners brukergrensesnitt med metoder som opprinnelig har blitt brukt ved design av brukergrensesnitt i PC/desktop-verdenen. De tre metodene er:

1. Brukbarhetstesting
2. GOMS-varianten Key-stroke Level Model (KLM)
3. Heuristisk evaluering

Vi vil forklare hver av de tre metodene i detalj senere i denne rapporten.

Hensikten med prosjektet er å finne ut om metodene kan brukes til å forbedre brukergrensesnitt på mobiltelefoner også. Når vi sier ”kan brukes til” legger vi flere ting i det; vi vil finne ut om metodene er gjennomførbare, effektive i bruk og om de gir resultater som kan føre til forbedringer av de mobile brukergrensesnittene.

### 2.2 Hvordan skal vi gjøre det?

Første steg i prosessen blir å bruke metodene i praksis. Vi vil evaluere brukergrensesnitt på tre forskjellige mobile enheter. På bakgrunn av den praktiske bruken av de tre metodene og data fra testene skal vi vurdere hvor godt metodene egner seg for evaluering av brukergrensesnitt på mobile enheter.

Trenton har laget et program som gjør det mulig å registrere hendelser (knappetrykk) ved bruk av en mobiltelefon. Programmet setter oss i stand til å registrere dataene vi trenger for å bruke KLM. Ettersom brukeren ikke vil kunne merke at vi registrerer knappetrykk, kan vi foreta brukbarhetstest og samle data til en Keystroke-Level Model samtidig. Heuristisk

evaluering av brukergrensesnittene vil bli en egen aktivitet vi gjennomfører etter at testene som involverer brukere er ferdige.

Oppdragsgiver Trenton stiller tre mobile terminaler til rådighet for oss. De tre terminalene vi skal evaluere er:

1. iPhone
2. Neo 1973 (OpenMoko)
3. Trolltech Greenphone

Terminal 2 og 3 kjører begge en applikasjonsplattform for mobile enheter som heter Qtopia Phone Edition. Begge telefonene har Linux-baserte operativsystemer.

## 2.3 Hva venter vi å finne?

Vi venter å finne at metodene egner seg til å evaluere mobile brukergrensesnitt. Vi tror utfordringen vil ligge i å beskrive hvor effektive de forskjellige metodene viser seg å være i forhold til hverandre og eventuelt hvilke tilpasninger som vil være hensiktsmessige.

## 3. Metodene

I det følgende vil vi se nærmere på hver av de tre modellene vi skal jobbe med, og forsøke å gi en forståelse av hvordan de fungerer.

### 3.1 Brukbarhetstesting

Denne formen for evaluering var den dominerende på 80-tallet, og er fortsatt viktig selv om andre former for evaluering har blitt mer og mer vanlige i den senere tid (Preece et al. 2002). Metoden karakteriseres av den sterke kontrollen arrangøren av testen har. Dette gjelder både for stedet testingen gjennomføres, som ofte er kontrollerte, laboratorielignende omgivelser, men også for hva den som testes kan foreta seg under testens forløp. Ofte sperrer det for innkommende samtaler på telefon, mobiltelefoner skrur av og personer som ikke deltar i testingen holdes borte. Under testingen registreres alt testdeltakerne foretar seg, slik at det senere kan gjennomgås. Metoden går ut på å observere og analyserer hvordan funksjoner i en løsning blir brukt av faktiske brukere. Man ønsker å kvantifisere brukernes ytelser, gjennom å identifisere og forstå problemområder eller komponenter som gjør bruken vanskeligere enn nødvendig, slik at man kan gjøre forbedringer på sluttproduktet.

#### 3.1.1 Brukbarhet

Brukbarheten av et brukergrensesnitt sier noe om hvor lett det er å lære seg bruken av grensesnittet, hvor effektivt det er og hvor godt brukeren liker grensesnittet. Man kan dele målene for brukbarhet opp i 6 forskjellige kategorier (Preece et al. 2002). *Effectiveness* er hvor godt grensesnittet fungerer til å løse oppgavene det er tiltenkt. *Efficiency* dreier seg om hvor effektivt grensesnittet utfører en oppgave, eller hvor godt tilrettelagt det er for brukere. *Safety* er også en viktig del av brukbarhetsmålene. Her dreier det seg om hvordan designet hindrer brukeren fra å komme opp i uønskede situasjoner, som for eksempel å slette en

kontakt eller en melding uten at man egentlig vil gjøre dette. *Utility* dreier seg om funksjonaliteten et design tilbyr en bruker. Har man de riktige verktøyene til å utføre en gitt oppgave? *Learnability* henviser til hvor enkelt det er å lære seg bruken av et design. Det siste delmålet er *memorability*, som tar for seg hvor enkelt det er å huske bruken av et brukergrensesnitt etter at man har lært seg dette en gang.

### 3.1.2 Komponenter i brukbarhetstesting

En brukertest består av ulike komponenter. Man har testdeltakere, et utvalg personer som brukes til å skape en forståelse av hvor godt et brukergrensesnitt fungerer. Videre trenger man et design man kan teste, med andre ord et brukergrensesnitt man ønsker å teste brukbarheten av. Man behøver også lokaler der testingen kan foregå, gjerne tilrettelagt for denne typen brukertesting. Ofte inneholder slike lokaler opptil flere kameraer, utstyr som registrerer tastetrykk og områder der arrangørene av testene kan overvåke uten å forstyrre testforløpet. Arrangørene av testen er også en viktig del. Disse kan fylle forskjellige roller, som for eksempel teknikker, prototype ekspert, passiv observatør eller teknikker.

### 3.1.3 Fordeler og ulemper

Denne formen for evaluering har, som andre evalueringsmetoder, både styrker og svakheter. Ser man på styrkene er det klart at en slik form for testing vil avdekke svakheter og problemområder på et tidlig stadie, og før produktet slippes på markedet. På denne måten kan produsenter sikre seg mot økonomiske tap som følge av problemer etter lansering. Videre vil produkter som tilfredsstillir brukeres krav til brukbarhet være med på å skape et godt rykte for produsenten. Man kan også anvende testresultatene som en referanse for fremtidige oppgraderinger av produktet, eller ved utviklingen av et nytt produkt. Ser man på svakhetene ved denne metoden, er det for det første ingenting som garanterer for at produktet aksepteres av brukere selv om man har gjort en brukbarhetstest. Antallet deltakere i en brukertest vil bare representere en liten andel av potensielle sluttbrukere, og denne formen for testing er ofte kostbar både økonomisk og tidsmessig. Det er også vanskelig å avdekke alle scenarioer et produkt kan komme opp i gjennom en brukbarhetstest.

## 3.2 GOMS

Tidlig på 80-tallet utviklet Stu Card, Tom Moran og Alan Newell GOMS-modellen (Card et al., 1983). Modellen tar for seg interaksjonen mellom menneske og maskin, og forsøker å modellere brukerens kunnskap og kognitive prosesser (Preece et al., 2002). Metoden tar for seg ekspertbrukeres omgang med brukergrensesnitt, det vil si brukere som kjenner programmet godt.

Selve navnet på modellen viser til de fire delene den inneholder. *Goals*, *Operators*, *Methods* og *Selection rules*. *Goals*, eller mål, er det brukeren ønsker å oppnå med handlingene sine. Dette kan være alt fra å finne frem til en side på internett, fjerne avsnitt i et tekstdokument eller å skrive ut et bilde. *Operators*, på norsk operatører, er de mentale og fysiske handlingene brukeren må gjennomgå for å komme til målet. Man bestemmer seg for eksempel for å bruke et program fremfor et annet til teksbehandling av et bestemt dokument, eller man bruker en nettbasert e-post klient fremfor maskinens installerte program. *Methods*, metoder, er sammensetningen av forskjellige operatører, som leder frem til målet. Man har foreksempel metoder for opprettelse av nye dokumenter i en teksteditor, og man har metoder for å gjøre

søk via søkemotorer. *Selection rules*, metodevalg, brukes dersom det er flere metoder som leder frem til et og samme resultat. Som et eksempel kan man tenke seg at man ønsker å gjøre et søk i en nettside etter ordet ”kake”. Man kan da enten benytte seg av menyen i toppen av skjermen, eller man kan gjøre bruk av de innebygde snarveiene i programmet. Ser man på Firefox som et eksempel, vil snarveien Ctrl+F, gi akkurat det samme resultatet som menyvalget Edit – Find in this page.

Modellen gir ingen tallfestede data som senere kan analyseres, og det ble derfor utviklet en ny metode å dokumentere brukeres utførelse av en gitt oppgave, keystroke level modell.

### 3.2.1 Keystroke Level Model

Også denne modellen ble utviklet av Card og hans team (Preece et al. 2002). Den store fordelene med denne modellen, er at den gir data i form av tall. På denne måten får man sammenlignet om en metode faktisk er raskere enn en annen. Det ble her fastsatt et sett med standardtider for de viktigste operatorene som benyttes i utførelsen av en oppgave. I tabellen under er tidene man kom frem til listet.

Operator	Beskrivelse	Tid (sekunder)
K	Trykke ned en enkelt knapp	0,35
P	Peke med et pekeverktøy	1,10
P1	Klikke med et pekeverktøy	0,20
H	Plassere hender på tastatur eller mus	0,40
M	Mental forberedelse	1,35
R(t)	Responstid fra systemet	t

Tabell 1 - KLM operatører med tider (Preece et al. 2002)

I senere tid er det kommet frem at man kan komme frem til et mer presist resultat gjennom å gjøre bruk av Fitts Law (Gong & Kieras, 1994). Fitts Law forutser tiden det tar å peke på et mål ved bruk av et pekerverktøy, basert på målets størrelse og avstanden pekeren må flyttes (Preece et al. 2002)

Operatorene bestemmes ut fra fysiske bevegelser hos en person, med unntak av de mentale forberedelsene, M, der det ble utviklet et sett med heuristikk for å plassere inn denne operatoren.

Ved å gå igjennom en metode, og bestemme de ulike operatorene en bruker vil nyttiggjøre seg av i utførelsen av denne, kan man regne seg frem til tiden det tar å utføre oppgaven ved bruk av formelen:

$$T_{\text{execute}} = T_K + T_P + T_H + T_D + T_M + T_{R(t)}$$

Selv om man får ut et kvantifiserbart resultat ved bruk av denne metoden, tar den ikke høyde for ulike brukernivåer. Det tas heller ikke høyde for ulike mental forberedelsestid mellom brukere. Man vil med andre ord ikke få et resultat som dekker både nybegynnere og ekspertbrukere. Ved bruk av denne metoden er det derfor viktig å legge de samme forutsetninger til grunn ved evaluering av for eksempel ulike brukergrensesnitt.

### 3.3 Heuristisk Evaluering

Heuristisk evaluering er en metode brukt for å avdekke feil og problemer i designet av brukersystemer. Prosessen innebærer at et knippe testpersoner går systematisk gjennom det aktuelle brukergrensesnittet og vurderer det opp mot ett gitt sett brukbarhetsprinsipper (heuristikker).

Vi velger å fokusere på det arbeidet som er utført av Jakob Nielsen. Hans erfaring innen feltet, kombinert med forskning han bedrev på 1990-tallet gjør ham til en autoritet innenfor emnet.

#### 3.3.1 Evalueringsprosessen

Heuristisk evaluering fordrer bruk av flere testpersoner av flere årsaker. Tidligere utførte studier (Nielsen, 1994) viser at en person vanligvis ikke vil avdekke alle problemer, og at forskjellige personer vil avsløre ulike problemer. Det blir allikevel hevdet at det er hensiktsmessig å begrense antall testpersoner til mellom 3 og 5, ettersom effektiviteten med flere testpersoner er drastisk avtagende.

De ulike testpersonene går igjennom i systemet på egenhånd. Testpersonen kan enten notere ned en rapport under gjennomgangen, eller det kan brukes en observatør som registrerer testpersonens kommentarer. Dersom testpersonen åpenbart sliter med systemet, har observatøren anledning til å komme med veiledning slik at man kan spare verdifull tid. I vårt tilfelle er det mest naturlig å bruke en observatør ettersom vi vil sikte på å ta i bruk utenforstående personer som testpersoner, og vi vil ikke belaste de med unødvendig arbeid.

I løpet av testen skal systemet gjennomgås minst et par ganger, og vurderes opp mot en liste av anerkjente brukbarhetsprinsipper. Anbefalt prosedyre er å benytte første gangen til å kjøre gjennom systemet for å få en følelse av hvordan ting fungerer, mens ved andre gjennomkjøring ligger fokus på brukbarhetsprinsippene.

Resultatene må forklares grundig og med spesielt henblikk på heuristikkene, det er ikke nok at testpersonen uttrykker misnøye med noe. Hvis det presenterer seg problemer relatert til flere heuristikker på ett element, må samtlige noteres.

#### 3.3.2 Fordeler med bruk av heuristisk evaluering

Heuristisk evaluering har flere fordeler ved seg. Først og fremst så er det en meget kostnadseffektiv måte å evaluere ett system på. Det er også en fordel at systemet man i utgangspunktet ikke er avhengig av å kjenne systemet i detalj. Metoden lar seg også abstrahere slik at den kan brukes på prototyper og prosjekter som ikke er realisert i sin tiltenkte tilstand. Dette gjør det mulig å implementere metoden gjennom store deler av utviklingsperioden, som igjen kan bidra til å eliminere feil på ett så tidlig tidspunkt som mulig.

#### 3.3.3 Nielsens heuristikker

Vi vil etter all sannsynlighet velge å ta utgangspunkt i heuristikkene som ble utviklet i samarbeid mellom Nielsen og Molich i 1990. Mer presist er dette en norsk oversettelse av Nielsens revisjon av dette arbeidet i 1994 (Nielsen, 1994).



- Synlighet av systemets status:
  - Systemet bør til enhver tid gi mulighet for å holde brukeren oppdatert på hva som foregår, gjennom logiske tilbakemeldinger innen rimelig tid.
- Korrelasjon mellom virkeligheten og systemet:
  - Systemet bør kommunisere på brukerens språk, og ikke i systemorientert lingo. Den bør følge konvensjoner fra den virkelige verden ved å gjøre informasjon tilgjengelig i logisk rekkefølge.
- Brukerkontroll og frihet
  - Når brukere gjør feil og havner i uønskede situasjoner, bør det være en enkel måte å komme seg ut av den på. Muligheten for å angre og omgjøre operasjoner bør vektlegges.
- Konsistens og standarder
  - Brukere skal ikke behøve å bekymre seg om hvorvidt ulike ord, situasjoner og utførelser betyr det samme i ulike settinger. Følg plattformspesifikke konvensjoner.
- Forebygg feil
  - Forsøk å forebygg feil i utgangspunktet. Fjern situasjoner som er tilbøyelige for feil, eller sørg for å gi brukerne en passende advarsel før de eventuelt begir seg ut på handlingen.
- Gjenkjennelse trumfer hukommelse
  - Minimer belastningen av brukerens hukommelse ved å gjøre objekter, handlinger og alternativer synlige. Brukeren må ikke burde huske informasjon fra en dialog til en annen. Instruksjoner for hvordan systemet fungerer burde til enhver tid være lett tilgjengelig.
- Fleksibilitet og effektiv bruk
  - Akseleratorer, som er usynlige for nybegynnerbrukeren, gjør systemet mer effektivt for erfarne brukere slik at systemet tar hensyn til begge type brukere. Brukere bør få tilrettelegge personlige innstillinger for ofte brukte handlinger.
- Estetisk og minimalistisk design
  - Dialoger bør ikke inneholde irrelevant eller lite brukt informasjon. All overflødig informasjon vil konkurrere med det relevante, og dermed gjøre det mindre synlig.
- Hjelp bruker å gjenkjenne, diagnosere og komme ut av feil
  - Feilmeldinger bør uttrykkes i ett kodefritt, enkelt språk som oppsummerer problemet på en oversiktlig måte og foreslår en konstruktiv løsning.
- Hjelp og dokumentasjon

- Optimalt sett bør ett system kunne brukes uten dokumentasjon, men det kan være nødvendig å tilby hjelp og dokumentasjon. All slik informasjon bør være enkel å søke gjennom, fokusert på brukerens oppgave, nevne konkrete steg som kan gjøres, og ikke være for omfattende.

## 4. Arbeidet hittil

Det første gruppa gjorde var å treffe oppdragsgiver Trenton Schulz. Trenton viste frem to av de mobile enhetene vi skal se nærmere på og ga en demo av programmet han har laget. Programmet registrerer knappetrykk/hendelser på de mobile enhetene slik at vi kan registrere data til bruk i KLM.

Da ingen av gruppemedlemmene var kjent med alle de tre metodene fra før, besto arbeidet videre i å lese seg opp på og sette seg inn i metodenes oppbygning og virkemåte. Vi hadde flere møter med Trenton hvor gruppa hadde anledning til å spørre om ting som var uklare og lære mer om metodene. Vi begynte etter hvert også planlegging av testene som vil involvere brukere og laget testplaner. Vi fikk låne to av enhetene, Greenphone og Neo 1973 (Openmoko) av Trenton slik at vi kunne sette oss inn i hvordan de fungerte.

I uke 10 (05.03.08) presenterte vi prosjektet og artikkelen ”When two methods are better than one: combining user study with cognitive modeling” i forelesningen i faget.

Status på prosjektet nå er at testingen er i gang og at den vil fortsette over påske. Vi satser på å gjennomføre brukbarhetstesting og KLM-logging.

## 5. Veien videre

Planen nå er å bruke metodene i praksis ved å gjennomføre testene. Første test blir fredag 14.03.07. Testene vil gi oss rådata vi kan arbeide videre med, samt gi oss innsikt i hvordan praktisk bruk av metodene arter seg. Etter oppsamlingen av dataene blir neste skritt tolkning av disse. Vi kommer til å ha kontrolltester som ”trial brukbarhetstest” (se artikkelen “When two methods are better than one: combining user study with cognitive modeling”) for å validere dataene og resultatene som vi får etter hvert.

Hele tiden skal det være samarbeid med vår gruppa og masterstudent Trenton Schulz ettersom han også trenger dataene vi får ut av testene i forbindelse med sin egen masteroppgave.

Arbeidet med testene og resultatene av disse vil forhåpentligvis gi oss svaret på vår problemstilling (se punkt 2) - om disse metodene kan brukes til mobile enheter. Om vi finner noen svakheter blir det vår oppgave å foreslå tilpasninger.

## 6. Referanser

- Card, S. K., Moran, T. P. & Newell, A. 1983, ”*The Psychology of Human Computer Interaction*” Lawrence Erlbaum Associates, USA.
- Preece, J., Rogers, Y. & Sharp, H. 2002, “*Interaction Design – beyond human-computer interaction*”, John Wiley & Sons, Inc., USA.
- Gong, R. & Kieras, D. 1994, “A validation of the GOMS model methodology in the development of a specialized, commercial software application”, [ACM](#) New York, NY, USA
- Nielsen, [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)
- Nielsen & Molich, [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)

## 7. Vedlegg

### 7.1 Artikkel: "When two methods are better than one: combining user study with cognitive modeling"

Artikkelen finnes på <http://portal.acm.org/citation.cfm?doid=1240866.1240900>

### 7.2 Testplan: Brukbarhetstest

Testplan for OpenMoko, Greenphone & iPhone

1. Ønsk testpersonen velkommen.
2. Informer om at vi på ingen måte er ute etter å teste dennes intellekt eller ytelse på andre måter.
3. Forklar at hensikten med testen er å forsøke å evaluere de mobile brukergrensesnittene, og at vi som test-team vil få mest ut av testen dersom testpersonen "tenker høyt" når oppgavene løses.
4. Informer testpersonen om at han/hun når som helst kan avbryte eller ta en pause hvis han/hun skulle føle behov for det.
5. Presenter testpersonen for selve oppgavene. Testpersonen begynner så snart han/hun er klar.
6. Husk å notere under testen. Hvis testen filmes må testpersonen informeres om dette og gi sitt samtykke.

- Fjern tastelås

- Legg til kontakten Marius med telefonnummer 45 00 24 28

- Send SMS meldingen "Hei" til nummeret 99 71 81 25

- Svar på innkommende samtale

- Avslutt samtalen

- Legg til nummeret fra forrige samtale til en ny kontakt, Jo Magnussen

- Finn kontakten Marius i adresseboken og ring

- Endre kontakten Jo til Jo Christian Magnussen

- Gjør telefonen lydløs

- Svar på meldingen fra Øyvind

- Sett på vekkerklokke for imorgen klokken 10:00

- Ta ett bilde og legg det til kontakten Øyvind

- Slå nummeret 45 00 24 28 og ring

- Ring siste oppringte kontakt
- Legg til kontakten Øyvind på "speed dial"
- Fjern bilde fra kontakten Øyvind
- Legg til "Møte på IFI" klokken 10:00 i kalenderen
- Sjekk tapte anrop
- Slett kontakten Jo Christian Magnussen
- Aktiver tastelås

Forutsetninger:

Før:

- Kontakten Øyvind må være tilstede
- Det må være en SMS fra Øyvind der
- Eksisterende tapte anrop

Etter:

- Slett kontakten "Marius"
- Slett "Møte på IFI" fra kalender