

# Sluttrapport

---

## Metoder for evaluering av mobile brukergrensesnitt

Jo Christian Magnussen, Marius Oppedal, Øyvind Reistad



# User interfaces



## Innholdsfortegnelse

1. Innledning .....	5
2. Prosjektet .....	5
2.1 Hva skal vi gjøre? .....	5
2.2 Hvordan skal vi gjøre det? .....	6
2.3 Hva venter vi å finne? .....	6
3. Metodene .....	6
3.1 Brukbarhetstesting .....	6
3.1.1 Brukbarhet .....	6
3.1.2 Komponenter i brukbarhetstesting.....	7
3.1.3 Fordeler og ulemper .....	7
3.2 GOMS.....	7
3.2.1 Keystroke Level Model .....	8
3.3 Heuristisk evaluering .....	9
3.3.1 Evalueringsprosessen .....	9
3.3.2 Fordeler med bruk av heuristisk evaluering.....	9
3.3.3 Nielsens heuristikker .....	9
4. Metodene i bruk.....	10
4.1 Brukbarhetstesting .....	10
4.1.1 Praktiske detaljer .....	11
4.1.2 Gjennomføringen .....	11
4.1.3 Resultater .....	11
4.2 GOMS/KLM.....	13
4.2.1 Testplan .....	13
4.2.2 Testsituasjonen .....	13
4.2.3 Resultater .....	13
4.3 Heuristisk evaluering .....	16
4.3.1 Valg av heuristikker .....	16
4.3.2 Gjennomføring .....	16
4.3.3 Resultater .....	16
5. Erfaringer fra bruk av metodene .....	21
5.1 Brukbarhetstesting .....	21
5.2 GOMS/KLM.....	21
5.3 Heuristisk evaluering .....	22

6. Konklusjon .....	23
6.1 Praktisk gjennomførbarhet .....	23
6.2 Styrker og svakheter .....	23
6.2.1 Brukbarhetstesting .....	23
6.2.2 Heuristisk evaluering .....	24
6.2.3 GOMS – Keystroke Level Model .....	25
7. Referanser .....	26
8. Vedlegg .....	27
8.1 Artikkel: “When two methods are better than one: combining user study with cognitive modeling” .....	27
8.2 Informed Consent Form: Brukbarhetstesting.....	27
Informed Consent Form .....	27
8.3 Testplan: Brukbarhetstest .....	28

## 1. Innledning

Dette dokumentet er sluttrapport for et studentprosjekt i faget INF5261 Utvikling av mobile informasjonssystemer ved UiO. Rapporten er ment å skulle:

- Definere prosjektoppgaven vi har jobbet med
- Gi leseren en innføring i evalueringsmetodene vi har vurdert
- Dokumentere den praktiske bruken av metodene og resultatene av disse
- Forsøke å komme til en konklusjon og gi et svar på problemstillingen definert i punkt 2 Prosjektet.

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 store utfordringer i å gjøre teknologiene i mobile enheter tilgjengelige og nyttige på brukernes premisser.

Det finnes metoder for evaluering av brukergrensesnitt på stasjonære enheter, og gruppen ønsker å undersøke om disse gir gode resultater også ved anvendelse på mobile enheter. Gjennom prosjektet håper gruppen å avdekke eventuelle svakheter og feil ved disse metodene, og på denne måten 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 teste evalueringsmetoder som brukes på stasjonære enheter, og finne ut om disse også er gyldige for bruk på mobile enheter. Gruppen vil i dette arbeidet ta for seg tre ulike metoder for evaluering. De tre metodene er:

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

Vi beskriver hver av de tre metodene nærmere under punkt 3 i denne rapporten.

Ved prosjektets slutt ønsker vi å kunne gi svar på om metodene er praktisk gjennomførbare og effektive samt hvilke styrker og svakheter vi fant ved hver av metodene. Vi vil eventuelt også foreslå tilpasninger eller endringer i metodene hvis det viser seg nødvendig.

## 2.2 Hvordan skal vi gjøre det?

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

Oppdragsgiver Trenton Schulz stiller tre mobile terminaler til rådighet for oss. De tre enhetene vi vil benytte i prosjektet 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 kjører operativsystemer basert på Linux-kjernen. iPhone kjører OS X operativsystem.

## 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, hvilke styrker og svakheter de har og å foreslå eventuelle tilpasninger som vil bedre metodene.

## 3. Metodene

I det følgende vil vi se nærmere på hver av de tre metodene vi skal jobbe med, og forsøke å gi en forståelse av hvordan de fungerer. Felles for evalueringsmetodene er at de alle er ment å skulle avdekke feil og mangler ved brukergrensesnittene man evaluerer.

### 3.1 Brukbarhetstesting

Den første metoden vi ser nærmere på var dominerende i evaluering av brukergrensesnitt 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 går ut på å observere og analysere hvordan funksjoner i et system faktisk blir brukt. På denne måten kan utviklere være sikre på at deres løsninger faktisk fungerer for brukere som ikke har deltatt i utviklingen. 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. Forstyrrende elementer, som personlige mobiltelefoner og annet, samt personer som ikke deltar i testingen holdes borte. Under testingen registreres alt testdeltakerne foretar seg ved hjelp av video, slik at materiale fra testen senere kan gjennomgås. Man ønsker å måle 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 brukbarhetstest består av ulike komponenter. Man har testdeltakere, et utvalg personer som brukes til å skape en forståelse av hvor godt et brukergrensesnitt fungerer. Lokaler der testingen kan foregå, gjerne tilrettelagt for denne typen brukertesting, er også en viktig bestanddel. 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 tekniker, prototypeekspert eller passiv observatør. Sist, men ikke minst er man avhengig av å ha et brukergrensesnitt man ønsker å teste.

### 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 kunne avdekke svakheter og problemområder på et tidlig stadium, og før produktet slippes på markedet. På denne måten kan produsentene av brukergrensesnittet redusere faren for økonomiske tap og andre problemer som følge av feil man oppdager etter lansering. Videre vil produkter som tilfredsstillter 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. I tillegg er denne formen for testing 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). Med kognitive prosesser menes prosesser som er knyttet til persepsjon og tenkning, altså bevisste prosesser. 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 tekstbehandling av et bestemt dokument, eller man bruker en nettbasert e-post klient fremfor

maskinenes 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 model.

### 3.2.1 Keystroke Level Model

Også denne modellen ble utviklet av Card og hans team (Preece et al. 2002). Den store fordelen med denne modellen er at den gir kvantitative 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 kvantitativt resultat ved bruk av denne metoden, tar den ikke høyde for ulike brukernivåer. Det tas heller ikke høyde for ulik 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

Den tredje metoden kalles heuristisk evaluering og er en metode brukt for å avdekke feil og problemer i designet av brukergrensesnitt. Prosessen innebærer at et knippe evaluatører 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 evaluatører 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 eksperter som evaluerer systemet til mellom 3 og 5, ettersom effektiviteten med flere evaluatører enn dette er drastisk avtagende.

Eksperten går igjennom i systemet på egenhånd. Han/hun kan enten notere ned en rapport under gjennomgangen, eller det kan brukes en observatør som registrerer ekspertens kommentarer. Dersom evaluatoren åpenbart sliter med systemet, har observatøren anledning til å komme med veiledning slik at man kan spare verdifull tid

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 hvor godt systemet stemmer overens med 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 er det en meget kostnadseffektiv måte å evaluere et system på. Det er også en fordel at 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 et så tidlig tidspunkt som mulig.

#### 3.3.3 Nielsens heuristikker

Vi har tatt utgangspunkt i heuristikkene som ble utviklet i samarbeid mellom Nielsen og Molich i 1990. Mer presist er dette en Nielsens revisjon av dette arbeidet i 1994 – oversatt til norsk. (Nielsen, 1994).

- **Visning av systemstatus**
  - Systemet bør til enhver tid gi mulighet for å holde brukeren oppdatert på hva som foregår, gjennom logiske tilbakemeldinger innen rimelig tid.

- **Samsvar 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.
- **Konsekventhet 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 forebygge 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 burde ikke måtte huske informasjon fra en dialog til en annen. Instruksjoner for hvordan systemet fungerer burde til enhver tid være lett tilgjengelig.
- **Fleksibilitet og effektivitet i 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 brukere å gjenkjenne, finne årsak til og komme ut feilsituasjoner**
  - 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 et 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. Metodene i bruk

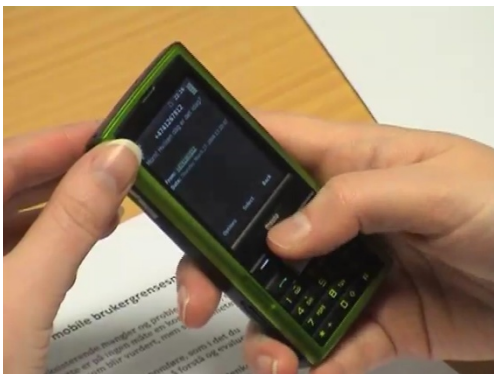
For å kunne avgjøre i hvilken grad metodene egnet seg for mobile brukergrensesnitt brukte vi alle tre i praksis. Vi testet hver av enhetene nevnt under punkt 2.2 med hver av de tre metodene i punkt 3.

### 4.1 Brukbarhetstesting

I dette avsnittet vil vi utdype rundt den praktiske gjennomføringen av brukbarhetstesting gjort på de tre telefonene.

#### 4.1.1 Praktiske detaljer

Brukbarhetstestingen ble gjort i et egnet rom, der gruppen kunne jobbe uforstyrret med testingen. Resultatene ble dokumentert gjennom bruk av videokamera, digitalt fotoapparat og notater. Testpersonene valgte vi ut gjennom å kontakte venner og bekjente, og blant personer uten kjennskap til de forskjellige telefonene. Antallet forsøk ble begrenset til to per enhet, dette fordi gruppen følte dette ville gi gode nok indikasjoner på hvordan testmetodene fungerte samt at denne formen for testing er svært ressurskrevende. Testpersonene ble gjort oppmerksomme på at seansen ble dokumentert med video og bilder til senere bruk, og gav samtykke til at dette kunne brukes på et senere tidspunkt i prosjektet. Bildene under viser henholdsvis screenshot fra video til venstre, og fotografi fra testseansen til høyre.



#### 4.1.2 Gjennomføringen

Testpersonene gjennomførte hver for seg et sett oppgaver med de ulike enhetene. Underveis i testene ble de oppfordret til å tenke høyt, og hele tiden fortelle hva de lette etter, hvorfor de valgte som de gjorde og om noe virket frustrerende eller lite forståelig. Dersom testpersonen stod fast i en oppgave, hjalp et av gruppemedlemmene til med kommentarer eller tips til hvordan problemet kunne løses.

#### 4.1.3 Resultater

Brukertestingen gav oss gode tilbakemeldinger på hvordan brukerne opplevde grensesnittene på de forskjellige telefonene. Selv om enkelte elementer ikke ble påpekt av samtlige testpersoner, var det stor enighet om gode og dårlige løsninger.

Den trykkfølsomme skjermen på OpenMoko telefonen var et av punktene der testpersonene var enige. Denne skapte vanskeligheter for brukeren da det ikke finnes merkede knapper for hverken mellomrom eller en egen knapp for fjerning av tekst. Dette hindret samtlige testpersoner i utførelsen av tildelte oppgaver. Også telefonens ordbok, som automatisk fullførte og rettet ord, ble påpekt som særdeles lite praktisk av testpersonene. Videre opplevdes det også som forvirrende å måtte trykke på knappen som ledet tilbake i menyen, back-knappen, for å lagre informasjon. Også at

brukergrensesnittet brukte lang tid på å bytte skjermbilder ved innkommende og utgående anrop var noe som skapte problemer ettersom brukere trykket flere ganger på svarknappen, og følgelig avsluttet samtalen. Av positive elementer ved dette brukergrensesnittet, trakk testpersonene frem et oversiktlig og godt menybilde.

Trolltechs Greenphone skapte også hodebry for testdeltakerne. Tekst skrevet inn av brukere måtte ofte bekreftes i to steg, noe som opplevdes som unødvendig og tungvint. Videre finnes det ingen mulighet til å gå tilbake og endre ord i tekstmeldinger, og ved opprettelse av nye telefonnummer i kontaktlisten hadde testpersonene problemer med å velge nummertype, som hjem, arbeid eller mobil, ettersom disse ikke er forklart i grensesnittet med annet enn ikoner. Brukerne oppfattet dette brukergrensesnittet som lite logisk og særdeles tungvint å bruke. Selv om telefonen støttet både navigasjon ved bruk av taster og navigasjon ved bruk av touch sensitiv skjerm, påpekte testpersonene at den berøringsfølsomme navigeringen ikke fungerte optimalt og at de heller foretrakk å benytte seg av tastene på telefonen.

Ser man derimot på Apples iPhone, som også benytter seg av touch-basert navigasjon, likte testpersonene denne atskillig bedre. Det er her tydelig at Apple har lagt mye jobb i å lage menyene og navigasjonen på denne telefonen så intuitiv og enkel som mulig. Allikevel påpekte brukerne at det hadde vært ønskelig med en egen knapp for å ringe kontakter, da det ikke var selvforklarende at oppringninger ble gjort ved å trykke direkte på kontaktens nummer. Også knappen som gjorde telefonen lydløs var problematisk å lokalisere. Det ble under testingen ytret tanker om at denne kunne vært merket tydeligere. Det var allikevel enighet blant testpersonene om at dette var enheten de foretrakk, og fant enklest å håndtere blant de tre som ble testet.

Personene brukte omtrent like lang tid på å fullføre settet med oppgaver for de tre telefonene, og det var ikke nødvendig med større inngrep fra de ansvarlige for testen for å sikre fremdrift hos brukerne. Enkelte hint og tips til løsninger på problemer ble allikevel gitt da brukerne fikk små problemer underveis i testene.

## 4.2 GOMS/KLM

Følgende avsnitt er en gjennomgang fra testen av de ulike enhetene ved hjelp av KeystrokeLevelModel.

### 4.2.1 Testplan

Basisen for testen er utviklingen av en logisk testplan for gjennomkjøring. For å ha en hensiktsmessig test, bør dette være relevante oppgaver som brukeren av enheten møter, og gjerne på jevnlig basis. I vårt tilfelle var det naturlig å bruke den samme testplanen som vi benyttet for brukbarhetstesting (se punkt 8.3).

### 4.2.2 Testsituasjonen

I vår testsituasjon er det to aktuelle personer som deltar. Den ene blir utstyrt med enheten og testplanen, som han skal gjennomføre punktvis som etter instruks fra observatøren. I vårt tilfelle var observatøren utstyrt med en Mac utstyrt med hjemmelaget software som avleste all input fra mobiltelefonene som var tilkoblet via en USB kabel. Observatørens oppgave var dermed å styre selve softwaren, som i hovedsak bestod i å starte og stoppe tidsmålingen av de ulike punktene i testplanen for brukeren.

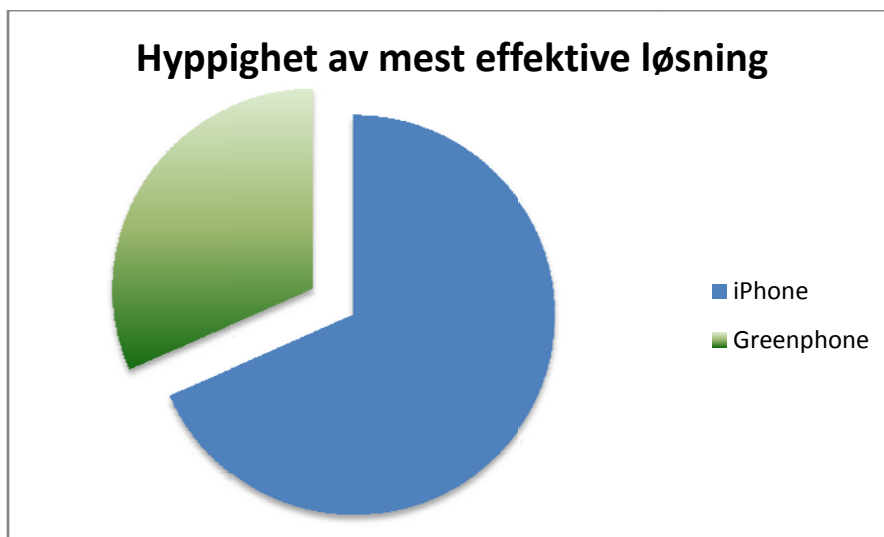
Softwareapplikasjonen beregner automatisk de ulike operatorene i KLM, (som nevnt under avsnitt 3.2.1) med oversikt over hvor lang tid som forløp på de respektive.

### 4.2.3 Resultater

Resultatene av KeystrokeLevelModel testingen står oppført i Figur 1. X-aksen benevner oppgavennummeret fra testplanen (se avsnitt 4.2.1), mens Y-aksen beskriver hvor lang tid det tok for de ulike enhetene fra start til slutt å utføre oppgaven. Søylefargen for iPhone identifiseres i blått, Greenphone i grønn og OpenMoko telefonen i fiolett.

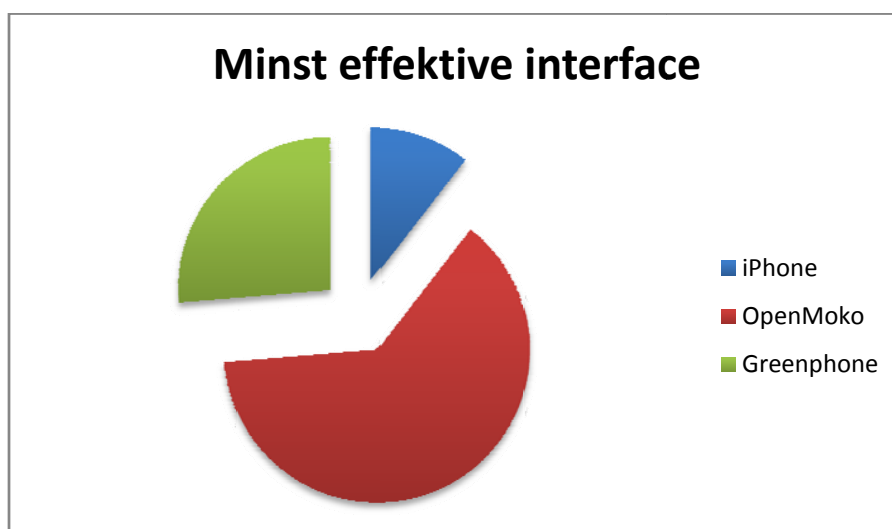
Av de konkrete resultatene som kan avleses i *Figur 3*, vil det være mulig å identifisere ineffektive arbeidsoppgaver for de enkelte systemene hver for seg. Ettersom vi nå har tilgjengelig materiale fra tre ulike enheter er det imidlertid interessant å sammenligne resultatene. Denne dataen kaster lys over svakheter og kan for eksempel brukes til å stille spørsmålstegn ved elementer av ett brukergrensesnitt. I vårt tilfelle kunne man for eksempel spørre seg hva det er som gjør at iPhone kun bruker 19150 millisekunder på å legge til et bilde til en kontakt under oppgave 12, mens OpenMoko bruker 43364ms, over dobbelt så lang tid.

*Figur 1 – Mest effektive interface*, beskriver antall ganger hvor de ulike systemene var kjappest til å utføre en oppgave. iPhone var desidert det mest effektive grensesnittet, med å være raskest til å utføre 65% av oppgavene. Greenphonen var i 30% av tilfellene, mens OpenMoko ikke var raskest en eneste gang.

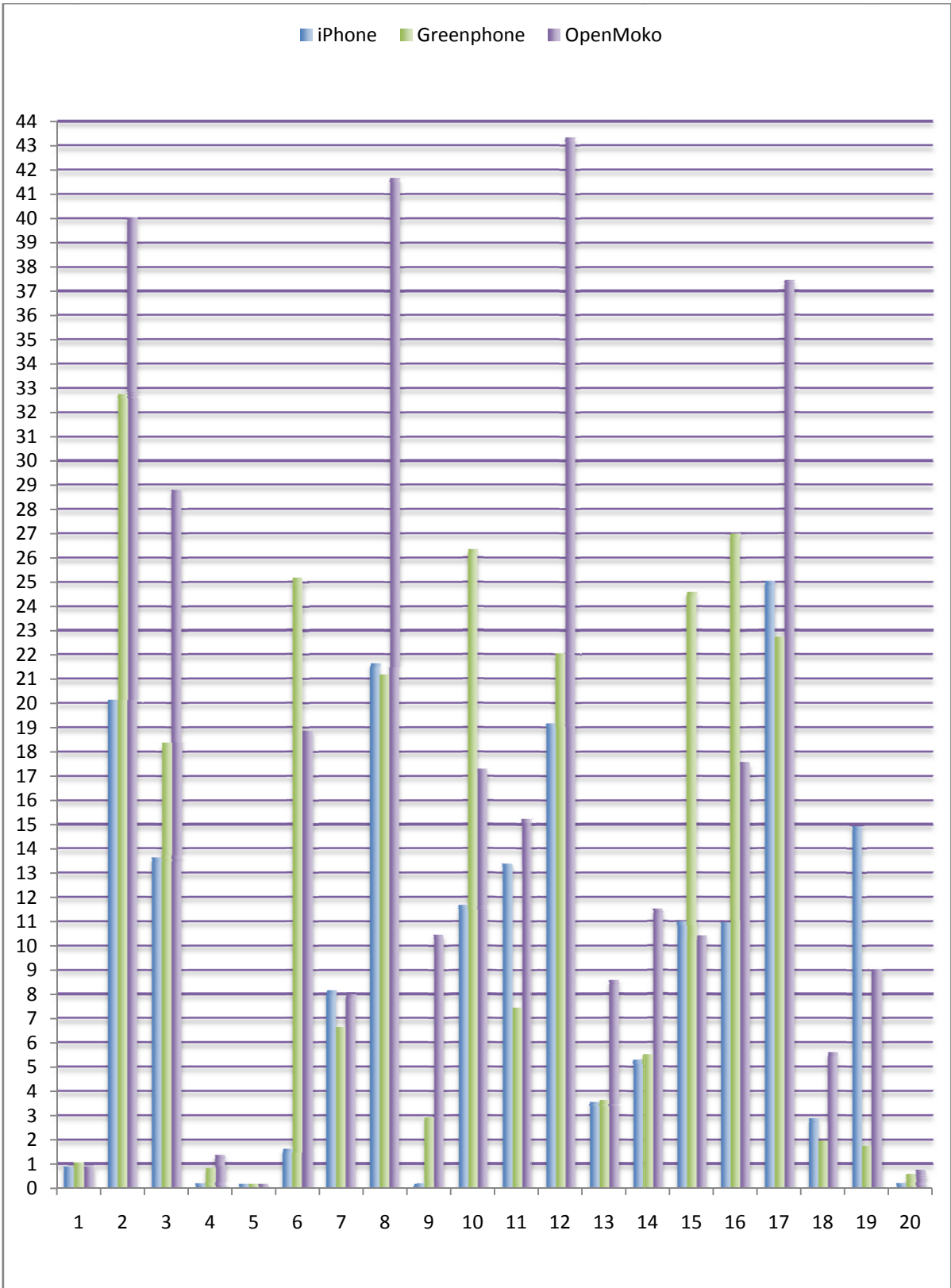


Figur 1 – Mest effektive interface

Figur 2 – Minst effektive interface, beskriver bildet av hvilket system som var treigest til å utføre de ulike oppgavene. Som vi kan lese av Figur 3, var OpenMoko treigest i 60% av alle oppgavene, Greenphone brukte lengst tid på 25% av oppgavene, mens iPhone var treigest på 10%.



Figur 2 – Minst effektive interface



Figur3 – X-aksen representer oppgavenummer i testplanen (avsnitt 4.2.1), mens Y-aksen antall sekunder passert for å utføre oppgaven

## 4.3 Heuristisk evaluering

I det følgende vil vi gå nærmere inn på hvordan metoden heuristisk evaluering fungerte i praksis.

### 4.3.1 Valg av heuristikker

Ved bruk av metoden heuristisk evaluering går en eller flere eksperter/evaluatorer gjennom et brukergrensesnitt for å evaluere hvor godt det aktuelle brukergrensesnittet følger reglene. Reglene er et sett heuristikker man velger ut på forhånd, men ikke desto mindre er de en sentral del av praktisk bruk av metoden. For denne evalueringen valgte vi Jakob Nielsens 10 heuristikker for brukergrensesnittedesign (Nielsen, 1994b).

### 4.3.2 Gjennomføring

Vi hadde behov for evaluatorene ved bruk av denne metoden, men følte at vi var skolerte nok til å klare dette selv og valgte derfor å ikke hente inn ekstern hjelp. Vi (gruppemedlemmene) slapp dermed å hente inn brukergrensesnitteksperter til å foreta evalueringen, og fylte heller denne rollen selv.

Selve den praktiske evalueringen foregikk ved at gruppemedlemmene først foretok en individuell evaluering av enhetene. Dette for å sikre rene og upåvirkede synspunkter fra evaluatorene (Nielsen 1994). Unntaket er Apple iPhone, da vi ikke hadde tilgang til denne enheten og derfor kun har tilgang til en heuristisk evaluering vår oppdragsgiver Trenton Schulz utførte i forbindelse med sin masteroppgave.

### 4.3.3 Resultater

I det følgende vil vi presentere punktene vi kom frem til etter å ha diskutert og satt sammen resultatene fra de heuristiske evalueringene.

#### 4.3.3.1 Greenphone

##### **Visning av systemstatus**

Enheter gir god tilbakemelding til bruker. Et eksempel på dette er at brukeren blir spurt om det er dette man virkelig ønsker når man forsøker å slette en kontakt. Et annet eksempel er at enheten gir tilbakemelding når man sletter meldinger ved å informere om at prosessen pågår og på den måten vise brukeren at det skjer noe.

Brukergrensesnittet gir tilbakemelding innen rimelig tid. Et eksempel er at når man sender tekstmelding gis det tilbakemelding innen rimelig tid om at meldingen er sendt. Nok et eksempel er at når man ringer til noen står det "Dialing.." i skjermen umiddelbart.

Generelt er det ikke så stort behov for å vise systemstatus da det er få operasjoner som tar lang tid slik at brukeren rekker å vente og lure på hva som skjer.

Det tar for lang tid å komme seg fra et menyvalg til et annet. Systemstatus blir da vist ved at en "klokke" vises på skjermen. Klokka signaliserer at enheten prosesserer. Den tida det tar må brukeren vente, og dette tar for lang tid!



## **Samsvar mellom systemet og den virkelige verden**

Samsvar med virkelig verden er god, med visse unntak. Vi har sett spesielt på "hovedmenyen", som er presentert med ikoner. De fleste ikonene svarer til ting i den virkelige verden på måter som brukerne sannsynligvis vil ha sett før og vil forstå. Noen av ikonene er vanskelige/umulige å forstå hva skal signalisere, for eksempel de fargede "boksene" som representerer menyvalget Applications.

Merk at hovedmenyen ikke viser tekst i tillegg til ikonene før man navigerer til ikonet med piltastene. Bruker man touchpad-funksjonaliteten må man gjette seg til hva ikonene betyr.

Ellers følger telefonen konvensjonene i mobilverdenen. Eksempel: kontaktliste, brev-ikon for SMS, hovedmeny.

## **Brukerkontroll og frihet**

Den røde telefonknappen bringer en alltid tilbake til forsidebildet/hovedskjermen. Dette er vanlig på mange telefoner (for eksempel Nokia). Brukeren har også alltid anledning til å benytte "Back"-knapp øverst til høyre for å gå ett steg tilbake, for eksempel i menyhierarkiet.

Vi fant imidlertid et sted i menyen hvor det er langt fra opplagt hvordan man skal gå ett steg tilbake. Når vi valgte "Documents" fra hovedmenyen og forsøkte å skrive inn bokstaver for å søke etter et dokument, var vi sterkt i tvil om hvordan vi skulle få fjernet bokstavene vi hadde skrevet inn og komme tilbake til utgangspunktet/den opprinnelige tilstanden. Situasjonen oppstår bare når man bruker den trykkfølsomme skjermen til å skrive inn søkestrengen.

## **Konsekventhet og standarder**

God på konsekvente formuleringer og standard. Eksempler er at man benytter "New" og "Delete" i forbindelse med tekstmeldinger og kontakter. Back-knappen vil også gi samme resultat hver gang, den vil bringe brukeren ett steg tilbake.

## **Forebygg feil**

Feilunngåelse: Vi så for oss et scenario hvor en bruker sletter en tekstmelding ved en feil. Det viser seg at sletteknappen for tekstmeldinger kalles "Move to trash". Gjør man denne operasjonen ved en feil, kan man hente seg inn igjen ved å gå til Trash og flytte meldinga tilbake til utgangspunktet.

## **Gjenkjennelse trumfer hukommelse**

Muligheter (options) er alltid lett tilgjengelig ved at linja nederst på skjermen viser tekst som signaliserer hva knappene gjør. Det er ikke nødvendig for brukeren å huske noe spesielt i bruken av grensesnittet.

## **Fleksibilitet og effektivitet i bruk**

Den har hurtigtaster i hovedmenyen ved at man kan trykke på ønsket menyvalg på keypad'en. Menyene lavere i hierarkiet har ikke denne funksjonaliteten.

En høyst aktuell snarvei for mobiltelefoner er jo "Ny SMS". Vi fant ikke en slik snarvei, og det så heller ikke ut til at det gikk an å opprette på en enkel måte. Samme gjelder for kontaktlista, man må gå via hovedmenyen. Tungvint.

### **Estetisk og minimalistisk design**

Vi fant ingen irrelevant informasjon i menyene.

### **Hjelp brukere å gjenkjenne, finne årsak til og komme ut av feilsituasjoner**

Fant ingen relevante feilmeldinger.

### **Hjelp og dokumentasjon**

Greenphone har kontekstavhengig hjelp. Man har "Help" som et eget menyvalg i alle "Options"-menyer i hele telefonen. Velger man hjelp, får man opp en tekst som beskriver menyen eller konteksten man er i.

#### **4.3.3.2 OpenMoko**

### **Visning av systemstatus**

Enheter viser systemstatus på en god måte. Man er aldri i tvil om hvilken oppgave man holder på med. Editerer man notater, vil man alltid ha overskriften "Notes".

### **Samsvar mellom systemet og den virkelige verden**

På dette punktet oppførte enheten seg på samme måte som Greenphone.

### **Brukerkontroll og frihet**

Brukeren har alltid back-knapp nederst i høyre hjørne. I og med at interaksjon med OpenMoko baserer seg på input via trykkfølsom skjerm, har man ingen knapper og dermed ikke den "globale" cancel-knappen som alltid bringer brukeren tilbake til utgangspunktet og på den måten alltid vil fungere som en vei ut for brukere som har kjørt seg fast.

### **Konsekventhet og standarder**

Her fant vi noen problemer. Når man oppretter en ny kontakt, ønsker man ofte å lagre denne. Problemet er at det finnes ingen lagreknapp, bare en knapp det står "Back" på. De fleste brukere vil assosiere denne knappen med at man avslutter oppgaven uten å lagre noe. Dette er ikke tilfelle, da Back-knappen i realiteten har lagrefunksjonalitet. Brukeren får heller ingen bekreftelse på at kontakten ble lagret, men må eventuelt sjekke dette manuelt ved å se i kontaktlista.

### **Forebygg feil**

Menyoppsettet signaliserer til tider feil funksjonalitet som kan føre til frustrasjon hos brukeren.

Et eksempel på dette er når man skal lagre en ny kontakt. Her er det lett å ta feil av knappen som gjemmer menyvalgene (hide), og knappen som avbryter hele operasjonen og returnerer brukeren til homescreen.

### **Gjenkjennelse trumfer hukommelse**

Ikonene benyttet i dette brukergrensesnittet er lett forståelige, og de fleste brukere vil kunne kjenne igjen, og assosiere disse med elementer fra hverdagen. Det er også her visse unntak. Enkelte menyelementer krever noe utforskning for å si sikkert hva de leder til.

Grensesnittet leverer gode tilbakemeldinger til bruker om hvor i systemet man befinner seg. Det er få støttefunksjoner i brukergrensesnittet, og enkelte elementer krever en del trening og tilvenning.

### **Fleksibilitet og effektivitet i bruk**

Det finnes en funksjon for snarveier, men denne viser seg å ofte bruke lenger tid på å komme til den valgte applikasjonen.

### **Estetisk og minimalistisk design**

Brukergrensesnittet viser sjelden irrelevant og unødig informasjon.

### **Hjelp brukere å gjenkjenne, finne årsak til og komme ut av feilsituasjoner**

Brukergrensesnittet trenger sjelden å vise feilmeldinger. Vi klarte ikke å gjenskape noen.

### **Hjelp og dokumentasjon**

Kontekstbaserte hjelpefunksjoner. Disse er lenket til andre elementer brukere kanskje ønsker å lære mer om i den aktuelle sammenhengen

#### **4.3.3.3 Apple iPhone**

De følgende resultatene er hentet fra en heuristisk evaluering foretatt av vår oppdragsgiver Trenton Schulz.

### **Visning av systemstatus**

På skjermen vises batterinivå, type internettilkobling (med styrke hvis WLAN), operatør og signalstyrke som konstante indikatorer.

Det vises også et tegn som indikerer hva som foregår hvis man laster ned data, har aktivert en alarm, tastelåsen står på eller enheten spiller av musikk. Skjermen viser ikke hvorvidt enheten er satt til lydløs eller ei, men dette ser man ved å snu på enheten og se hvilken stilling bryteren for lyd på/av står i. Enheten vil indikere hva som skjer hvis man ringer eller sender meldinger med den.

Det finnes tilfeller da en iPhone ikke vil gi respons, for eksempel vil det oppstå en forsinkelse når man forsøker å hente fram et notat man forlot i endringsmodus. Brukeren vil ikke få noen tilbakemelding mens forsinkelsen pågår.

### **Samsvar mellom systemet og den virkelige verden**

iPhone har svært konsistent bruk av vokabular i menysystemene, og uttrykkene er velkjent sjargong i mobilsammenheng. Det eneste punktet hvor det oppsto tvil var i beskrivelsen av noen uttrykk i terminalens funksjonalitet for å vise bilder i slideshow.

### **Brukerkontroll og frihet**

Brukeren vil alltid kunne trykke "Home"-tasten som en nødutgang fra enhver situasjon. Når man starter en applikasjon man tidligere har jobbet med, husker de fleste applikasjoner hvor de var og hva de drev med. Det eneste tilfellet hvor man potensielt kan miste data er når man legger inn info tilhørende en ny kontakt i kontaktlista. Hopper man tilbake til startskjermen mens man gjør dette vil dataene bli borte. Telefonen innehar ingen form for angre- eller angre tilbake-funksjonalitet.

## **Konsekventhet og standarder**

iPhone har konsekvente menyer og virker på samme måte gjennom hele menysystemet. Eneste å sette fingeren på er at enkelte applikasjoner snudde på seg på skjermen når man snudde telefonen mens andre applikasjoner igjen (som man skulle tro hadde denne funksjonaliteten) lot være.

## **Forebygg feil**

Det er ved inntasting av tekst at det oppstår flest feil. iPhone forsøker å komme med alternative forslag til og rette på det som blir skrevet, med det utfall at telefonen fra tid til annen ender opp med å rette ord og uttrykk man i utgangspunktet ikke skrev feil. Et annet potensial for feilsituasjoner ligger i telefonens samtalehistorikk. Feilen som kan oppstå her er at man ringer et nummer i historikken når man forsøker å se samtaledetaljer for en oppføring. Det som er positivt er at telefonen gir klar beskjed om at den ringer og at man således kan avbryte oppringningen. Et eksempel på feilunngåelse finner man ved sending av SMS eller e-post. I lista over kontakter man kan sende til, vil det indikeres hvis kontakten ikke er lagret med telefonnummer eller eventuell e-postadresse.

## **Gjenkjennelse trumfer hukommelse**

iPhone har store ikoner med tekst under hvert ikon som gjør det lett å huske hva applikasjonene gjør. Det er heller ikke nødvendig å huske telefonnumre eller e-postadresser, da telefonen foreslår mottaker fra kontaktlista når man begynner å skrive kontaktens fornavn ved sending av SMS eller e-post. Som regel finner den det man søker, selv om man ikke har skrevet helt rett. Unntaket er når man skal bruke kontaktlista for å finne en kontakt man ønsker å ringe til, men husker man de første bokstavene i kontaktens etternavn kan man benytte det til å snevre inn søket.

## **Fleksibilitet og effektivitet i bruk**

Det finnes ingen snarveier/akselleratorer i tradisjonell forstand. Det er mulig å begynne å skrive inn tekst istedenfor å måtte lete igjennom ei lang liste med oppføringer. Som nevnt i den forrige heuristikken, går det an å taste inn de første bokstavene i etternavnet til en kontakt istedenfor å måtte scrolle gjennom ei lang liste. Den viktigste formen for akselerator i enhetens brukergrensesnitt lagrer valg man foretar ofte. For eksempel kan man ha favoritter i kontaktlista, websider kan ha bokmerker og kartprogrammene kan ha lagrede lokasjoner. De nyeste SMS-samtalene blir også lagret og er lett å hente frem igjen. Det finnes ingen klipp og lim-funksjonalitet.

## **Estetisk og minimalistisk design**

Alle dialogene i iPhone følger denne heuristikken godt. Informasjonen blir stort sett presentert på en måte som legger til rette for touchscreen-funksjonaliteten.

## **Hjelp brukere å gjenkjenne, finne årsak til og komme ut av feilsituasjoner**

Man støter på svært få feilmeldinger ved normal bruk. De eneste vi oppdaget var melding om lavt batterinivå og beskjed om at SIM-kort mangler. Disse feilmeldingene spretter opp og forklarer situasjonen. Enheten gir ingen foreslag til hvordan man løser problemene, men løsningen burde være relativt åpenbar for de fleste.

## Hjelp og dokumentasjon

Det er ingen online hjelpefunksjon inkludert i selve iPhone. Det er mulig å navigere til Apples hjemmeside og få hjelp der, men det er ikke gitt at man har aksess (og det kan koste penger). De færreste operasjoner krever veiledning, men utfordringer som å avslutte en applikasjon som har krasjet eller restarte enheten har ikke åpenbare løsninger.

## 5. Erfaringer fra bruk av metodene

I dette kapittelet vil vi se nærmere på erfaringer vi gjorde oss ved bruk av metodene.

### 5.1 Brukbarhetstesting

Etter å ha gjennomført brukbarhetstesting på de tre telefonene, er det klart at denne metoden vil svakheter i brukergrensesnittets design også på mobile enheter. Metoden har mange fordeler, deriblant at man får testet sitt produkt på faktiske brukere. På denne måten kan man unngå problemene som kan oppstå dersom man ser seg blind på sine egne løsninger i en utviklingsfase.

Men det finnes også negative aspekter ved denne metoden. Først og fremst er metoden svært ressurskrevende. Dette gjelder både for innsamlingen av data, men også for analyse av disse på et senere tidspunkt. Mye tid gikk med til å gjennomgå videoopptak og analysere disse etter brukertesting.

Gjennom vår bruk av metoden oppdaget vi elementer som bør implementeres dersom man skal ha fullt utbytte av denne på mobile enheter. Først og fremst bør man inkludere en kobling mellom enheten som testet og en datamaskin som kan loggføre alle handlinger som utføres på testenheten. Dette var noe vi savnet etter å ha gått igjennom videoopptakene fra testene, der det til tider er vanskelig å få med seg hva som skjer på enheten som testes. Videre er det ønskelig med flere testpersoner enn vi benyttet oss av dersom man skal sikre et bredt resultat, der flere problemområder avdekkes. Det vil også være å foretrekke at settet med oppgaver som skal gjennomføres skreddersys den enkelte enhet, for på denne måten å slippe øyeblikk der brukeren står fast som følge av en oppgave som ikke lar seg utføre.

### 5.2 GOMS/KLM

Etter å ha gjennomført 3 vellykkede tester med KeystrokeLevelModel, vil vi konkludere med at dette er en metode som er velegnet for testing av mobile terminaler.

Forsøkene avslørte imidlertid et par faktorer som er interessante og verdt å ta stilling til eventuelt før man velger å benytte en slik form for evaluering.

Bruken av dataverktøy vil være bortimot obligatorisk. Muligheten til å ta tiden på ulik input fra enhetene for å få ett nøyaktig evalueringsmateriale er signifikant. Det ville vært ekstremt vanskelig, om ikke umulig, å oppnå tilsvarende detaljerte resultater dersom en skulle ha gjort ett slikt forsøk manuelt. I tillegg ville en slik tilnærming til metoden vært ekstremt mye mer tidkrevende. Dette vil nok fungere som en barriere for slik testing av mobile enheter, ettersom det ikke finnes standardiserte programmer til hjelp for nettopp dette.

En annen faktor som er verdt å vurdere, er verdien av å ha materiale fra tilsvarende enheter til å sammenligne med. Eksempelvis hadde vi to telefoner som benyttet seg av såkalt touchscreen

teknologi, og det var som vi har sett nokså stor forskjell på effektivitetsnivået i manøvreringen av brukersystemene. Fra vår erfaring har dette nødvendigvis ikke bare med selve oppsettet av interfacet å gjøre. Det kan også si noe om hvor godt andre aspekter ved telefonen fungerer, som for eksempel at touchscreenen fungerer bedre ved en enhet kontra en annen.

### 5.3 Heuristisk evaluering

Vår erfaring med heuristisk evaluering er at den i alle høyeste grad er gjennomførbar for mobile enheter. Det er ingenting som hindrer en brukergrensesnitteksperter i å gå igjennom et brukergrensesnitt på en mobil terminal med den hensikt å avdekke avvik fra heuristikkene. Vi synes også metoden er hensiktsmessig i og med at den har følgende fordeler:

- Enkel å gjennomføre
- Krever ingen andre forberedelser enn valg av heuristikker
- Ikke veldig tidkrevende
- Egnet til å avsløre feil ved mobile brukergrensesnitt

Grunnen til at vi vurderer metoden til å være egnet til å avsløre feil ved mobile brukergrensesnitt er at den gjorde det i vår praktiske utprøving. Eksempler på feil vi avslørte var for eksempel i Greenphones hovedmeny, hvor ikke alle ikonene samsvarer med den virkelige verden slik at det er mulig å forstå dem uten forklarende tekst. Et eksempel er ikonet for applikasjoner, som er illustrert med esker i fire forskjellige farger. Det er ikke åpenbart at dette menyvalget fører brukeren til "applikasjoner". Vår vurdering støttes av testpersonene i brukbarhetstesting, som kommenterte nøyaktig samme designproblem når de ble spurt om de forsto hovedmenyen.

Et annet eksempel er mangelen på lagreknapp ved opprettelse av nye kontakter i OpenMoko-enheten. Istedenfor å kunne trykke lagre og kanskje til og med få en bekreftelse, må brukerne trykke "Back" for å lagre en ny kontakt. Dette er ikke særlig god brukergrensesnittdesign, og bryter med Nielsens heuristikker (systemstatus kommer ikke fram tydelig nok – brukere vil spørre seg om den nye kontakten ble lagret?). Ønsker brukeren bekreftelse på at kontakten faktisk er lagret, må han/hun manuelt finne den fram i kontaktlista.

Denne feilen er notert som en designsvakheter i den heuristiske evalueringen vår, og ble også kommentert av testpersonene under brukbarhetstesting, siden det å lagre en ny kontakt inngikk i oppgavesettet disse skulle utføre.

Vi la under vår heuristiske evaluering godt merke til at heuristikkene vi hadde valgt ikke var tilpasset mobile enheter. Vi opplevde for eksempel punktet "Feilsituasjoner" som mindre aktuelt, da det så ut til at det sjelden oppsto slike situasjoner.

Et av våre forslag til forbedringer ved denne metoden er derfor at heuristikkene må velges med omhu – finner man heuristikker som sier noe om hvordan et godt, mobilt brukergrensesnitt skal være og ellers er tilpasset mobile enheter, vil det være det beste. Dette ga oss nok en påminnelse om at man på ingen måte kan sette likhetstegn mellom brukergrensesnitt for mobile og stasjonære enheter.

En annen ting man skal være obs på er antall evaluatorene. Nielsen anbefaler mellom 3 og 5 evaluatorene (Nielsen, 1994). Bruk av flere evaluatorene enn dette har vist seg å gi liten uttelling i forhold til investeringen. I vår heuristiske evaluering av OpenMoko og Greenphone ble det benyttet 2

evaluatorene, ved evaluering av iPhone kun én. Uten at vi har foretatt samme heuristiske evaluering med flere evaluatorene slik at vi kan sammenlikne, ser vi for oss at vi kanskje hadde funnet flere feil ved bruk av tre evaluatorene. Vi vil derfor foreslå at Nielsens anbefaling om 3-5 evaluatorene også følges ved evaluering av mobile brukergrensesnitt.

## 6. Konklusjon

Etter å ha testet de tre evalueringemetodene i praksis sitter vi igjen med kunnskap om hvor godt evalueringemetodene i punkt 3 egnede seg til evaluering av mobile brukergrensesnitt. Vi vil i denne konklusjonen forsøke å gi et kort resymé av hva vi har kommet fram til.

### 6.1 Praktisk gjennomførbarhet

Det første vi ønsket å få svar på var om metodene var praktisk gjennomførbare. Vi mener at svaret på dette er ja. Bortsett fra at skjerm-, tastatur- og enhetsstørrelse for mobiltelefoner generelt er en god del mindre enn for tradisjonell PC, var det praktisk mulig å bruke alle de tre evalueringemetodene:

- For den heuristiske evalueringens del gikk gjennomføringen ut på å bruke enhetene og kontrollere at de stemte overens med heuristikkene. Det er ingenting ved det mobile formatet som hindrer dette.
- Når det gjelder GOMS – KLM benyttet vi oppdragsgiver Trenton Schulz' programvare til å registrere hendelser og måle tid. Dette gjorde det praktisk mulig å benytte metoden ved at mobiltelefonen var koblet til en PC og programvaren registrerte dataene.
- Brukbarhetstesting ble gjennomført ved at testpersoner løste et sett oppgaver ved hjelp av brukergrensesnittet. Vi fant ingen egenskap ved det mobile formatet som umuliggjorde dette.

Det er imidlertid klart for oss at en kombinasjon av de tre metodene vil kunne gi et atskillig bedre resultat enn om metodene anvendes enkeltvis.

### 6.2 Styrker og svakheter

Vi ønsket også å se nærmere på styrker og svakheter ved metodene i den aktuelle konteksten. I det følgende vil vi kort oppsummere hva vi kom frem til. I drøftingen av eventuelle svakheter vil vi også komme med forslag til utbedringer.

#### 6.2.1 Brukbarhetstesting

Vi mener at en av de største fordelene ved metoden er at virkelige brukere tester brukergrensesnittet, og at dette kan være en kraftig syretest på hvor brukbart designet er. Dette kjenner vi til fra bruk av metoden i desktop-verden, men vi opplevde at metoden også var egnet til å avdekke designsvakheter ved mobile brukergrensesnitt.

Et poeng som er verdt å merke seg er at oppgaven testpersonene skal gå igjennom bør tilpasses nøye når man skal teste mobile brukergrensesnitt. Testteamet må stille seg spørsmål som:

- Hva er de vanligste og oftest brukte handlinger og verktøy?
- Hvilken kontekst er det mest vanlig at brukeren befinner seg i når han/hun bruker disse handlingene?

Et tungvint, mobilt brukergrensesnitt vil ikke nødvendigvis bli avslørt av en brukbarhetstest hvis ikke oppgavene testpersonene får er nøye tilpasset. Man lar ofte testpersonene i en brukbarhetstest på et tidspunkt få komme fritt til orde med en subjektiv vurdering av brukergrensesnittet. Siden de fleste testpersoner gjerne vil ha noe erfaring med bruk av mobile brukergrensesnitt fra før, tror vi det vil være en god idé å legge mye vekt på uttalelser som kommer fram i slike vurderinger.

Ulempen med metoden brukbarhetstesting er investeringen metoden krever i form av tid og arbeid. Som nevnt må man utarbeide oppgaver til testpersonene på forhånd. Disse må også prøves ut på forhånd for å unngå problemer og å sikre at gjennomføringen av testene flyter godt. Vi anbefaler å gjennomføre en pilottest, som best kan beskrives som en generalprøve før selve brukbarhetstesting.

I tillegg må en brukbarhetstest dokumenteres godt hvis man ønsker å skaffe seg mest mulig materiale for senere å kunne gjennomføre grundig analyse. Vi benyttet skriftlige notater og filmopptak, og opplevde at filmopptaket var det mest verdifulle materialet for oss i ettertid.

Mange mobile enheter har små skjermer. På grunn av dette vil det iblant være vanskelig for andre enn testpersonen å følge med på hva som skjer på skjermen under brukbarhetstesting. I vår test sto kameraet på et stativ, og vi zoomet inn på mobiltelefonens skjerm. Vi løste problemet med å høflig, men bestemt be testpersonen ikke være i veien for kameraet. Allikevel ser vi for oss at den ideelle løsningen hadde vært å ha direkteoverføring av mobiltelefonens skjerm til en PC med større skjerm. Hvis man i tillegg kunne gjøre opptak av skjermbildet i tillegg til å registrere tastaturhendelser på mobiltelefonen, ville man få et enda bedre utgangspunkt for analyse.

Navigasjonen på mobile enheter er ekstremt viktig, og det er derfor vår mening at dette bør være i fokus når mobile enheters grensesnitt testes.

### 6.2.2 Heuristisk evaluering

Vi opplevde at denne metoden var rask å bruke, og ser på dette som et av metodens hovedfortrinn. Forberedelsene består av valg av heuristikker samt innhenting av brukergrensesnitteksperter til å foreta selve evalueringen. Siden vi evaluerte selv, unngikk vi sistnevnte forberedelse.

Erfaringen vi gjorde oss var at metodens egnethet til en viss grad ble bestemt ut fra heuristikkene. Vår vurdering er at metodens evne til å avdekke designsvakheter avhenger litt av de heuristikkene man velger. Hvis man velger for generelle heuristikker risikerer man å overse enkelte designfeil eller svakheter. I tillegg risikerer man i en slik evaluering å fremstille et mobilt brukergrensesnitt positivt, enda det fungerer dårlig i enhetens tiltenkte kontekst. Vi opplevde for eksempel at deler av både Greenphone og Neo1973's brukergrensesnitt ble mislikt av testpersonene i brukbarhetstesten, mens samme designfeil kom mindre tydelig fram i den heuristiske evalueringen.

Vårt forslag til forbedring blir derfor å tilpasse heuristikkene man bruker slik at de tar hensyn til kontekst og tiltenkt brukssituasjon. Det er ikke vanskelig å se for seg at for eksempel en mobiltelefon med mp3-avspillingsfunksjonalitet bør ha heuristikker med litt andre fokusområder enn en mobil GPS.



### 6.2.3 GOMS – Keystroke Level Model

KeystrokeLevelModel er en metode som lar seg bruke på mobile terminaler. Men som vi har sett er det en utfordring å benytte slike tester uten dataverktøy, og det ligger utfordringer i å fremstille slikt i mangel på standardiserte hjelpemidler.

Avkastningen av en slik evaluering kan imidlertid være stor.

For det første, dersom man ser vekk i fra eventuelle utviklingskostnader av evalueringsverktøy, så er det en rimelig evalueringsmetode som ikke krever store mengder ressurser for å gjennomføres, verken i form av penger, tid eller antall personer.

For det andre kan det være med på å avsløre lite logiske og lite gjennomtenkte brukergrensesnitt. Ved å sette systemet i slikt perspektiv kan man også skape oversikt som kan inspirere til nye tanker angående logisk rekkefølge og stimulere til nye snarveier innenfor grensesnittet.

For det tredje, nærmest som en bieffekt kan det være med på å kaste lys over andre problemområder. Som vi var inne på tidligere med tanke på OpenMoko sitt touchscreen-system, kan testene også underbygge at inputmetoder, altså ting som er mer hardware relaterte, ikke fungerer optimalt.

## 7. Referanser

1. Card, S. K., Moran, T. P. & Newell, A. 1983, "*The Psychology of Human Computer Interaction*" Lawrence Erlbaum Associates, USA.
2. 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
3. Nielsen, Jakob, 1994: Heuristic evaluation  
[http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)
4. Nielsen, Jakob, 1994b: Ten general principles for user interface design,  
[http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)
5. Preece, J., Rogers, Y. & Sharp, H. 2002, "*Interaction Design – beyond human-computer interaction*", John Wiley & Sons, Inc., USA.

## 8. Vedlegg

### 8.1 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>

### 8.2 Informed Consent Form: Brukbarhetstesting

#### Informed Consent Form

I state that I am over 18 years of age and wish to participate in a program of research being conducted by Daniel Lagutin, Jo Christian Magnussen, Marius Oppedal, Øyvind Reistad at the University of Oslo, Department of Informatics.

The purpose of the research is to assess the usability of three mobile phones: the Qtopia Greenphone, the Neo1973, and the iPhone.

The procedures involve the monitored use of the phones. I will be asked to perform specific tasks using the phones. I will be asked open-ended questions about the phones and my experience using it.

All information collected in the study is confidential, and my name will not be identified at any time.

I understand that I am free to ask questions or to withdraw from participation at any time without penalty.

\_\_\_\_\_  
Signature of Participant                      Date                      Place

\_\_\_\_\_  
INF5261 Group Member                      Date                      Place

### 8.3 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