



Tuomo Saunakangas

GRAAFISEN KÄYTTÖLIITTYMÄN SUUNNITTELU ANDROID- POHJAISEEN LAITTEESEEN

GRAAFISEN KÄYTTÖLIITTYMÄN SUUNNITTELU ANDROID- POHJAISEEN LAITTEESEEN

Tuomo Saunakangas

Opinnäytetyö

Kevät 2012

Tietotekniikan koulutusohjelma

Oulun seudun ammattikorkeakoulu

| | | | | |
|-------------------------|--|--------|---|-----------|
| Koulutusohjelma | Opinnäytetyö | Sivuja | + | Liitteitä |
| Tietotekniikka | Insinööriö | 44 | + | 0 |
| Suuntautumisvaihtoehto | Aika | | | |
| Sulautetut järjestelmät | Kevät 2012 | | | |
| Työn tilaaja | Työn tekijä | | | |
| Navicron Oy | Tuomo Saunakangas | | | |
| Työn nimi | Graafisen käyttöliittymän suunnittelu Android-pohjaiseen laitteeseen | | | |
| Avainsanat | Android, käytettävyys, käyttöliittymäsuunnittelu, käyttöliittymät | | | |

Insinööriössä suunniteltiin graafinen käyttöliittymä Android-pohjaiseen laitteeseen. Työn toimeksiantajana oli Navicron Oy, joka toimii Oulun teknologiakylässä. Navicron Oy kehittää asiakkaidensa tarpeiden mukaisia langattoman teknologian tuotteita.

Työn tavoitteena oli tehdä asiakkaan tilauksen mukainen graafisen käyttöliittymän demo sekä saada se toimimaan Freescalen i.MX53-alustalla. Työn ohella oli tarkoitus oppia käyttöliittymäsuunnittelun perusasioita sekä tutustua suunnittelussa käytettäviin työkaluihin.

Työn tietoperustana sovellettiin pääasiassa Mentor Graphicsin Inflexion UI –training-sivuston sisältämiä videoita sekä muita oppimateriaaleja. Jotta suunnittelutyössä kyettiin käyttämään Inflexion UI:tä, oli harjoitettava ohjelman käyttöä. Opinnäytetyön teoriaosuuden lähdeaineistona käytettiin kirjoja ja verkkomateriaalia. Ajantasalla olevaa painettua materiaalia ei ollut juurikaan saatavissa, joten piti tukeutua sähköiseen materiaaliin.

Projektin tuloksena saatiin kehitettyä asiakkaan tilauksen mukainen demo, joka voi toimia pohjana myöhemmin mahdollisesti toteutettavalle laitteelle, mikäli asiakas niin haluaa. Myös käyttöliittymäsuunnittelu sekä siihen käytetyt työkalut tulivat tutuiksi. Arvioitaessa työn tuloksia, voidaan todeta, että jos laite toteutetaan, kannattaa sitä suunniteltaessa käyttää toisia työkaluja.

| | | | | |
|--|-------------------|-----------------|---|------------|
| Degree programme | Thesis | Number of pages | + | appendices |
| Information Technology | B.Sc. | 44 | + | 0 |
| Line | Date | | | |
| Embedded systems | Spring 2012 | | | |
| Commissioned by | Author | | | |
| Navicron Ltd | Tuomo Saunakangas | | | |
| Thesis title | | | | |
| Designing a graphical user interface for an Android-based device | | | | |
| Keywords | | | | |
| Android, usability, user interface design, user interfaces | | | | |

In this thesis a graphical user interface for an Android-based device was designed. The client of the work was Navicron Ltd. Located in the Oulu technology village. Navicron Ltd. develops wireless technological products for customers needs.

The objective of this thesis was to design a graphical user interface demo and make it work on Freescale i.MX53 device, as requested by the customer. Another goal for the work was to learn the basics of ser interface design and to get familiar with designing tools.

The knowledge base of the work was mainly Mentor Graphics Inflexion UI training webpage, videos and other training materials. In order to be able to use Inflexion UI in designing, it was first necessary to get familiar with the software in question. Books and web-based materials were used as the source of theoretical part of the thesis. As there was hardly any up to date printed material available, this thesis relies mostly on digital sources.

As a result of the project, a demo fit to the customer's order was developed. The demo can work as a base to probably later on production if the customer so wants. UI designing and the tools that were practiced became also familiar. When estimating the results of the work I was wondering if some other tools were better when possible planning a device that is feasible.

SISÄLTÖ

| | |
|---|----|
| TIIVISTELMÄ | 3 |
| ABSTRACT | 4 |
| SISÄLTÖ | 5 |
| LYHENTEET JA TERMISTÖ..... | 7 |
| 1 JOHDANTO..... | 9 |
| 2 KÄYTTÖLIITTYMÄN PERUSTEET | 10 |
| 2.1 Käyttöliittymän tärkeys | 10 |
| 2.2 Komentoliittymä | 11 |
| 2.3 Tekstipohjainen käyttöliittymä | 12 |
| 2.4 Graafinen käyttöliittymä | 13 |
| 2.5 Käyttöliittymät tulevaisuudessa | 14 |
| 3 KÄYTETTÄVYYS | 15 |
| 3.1 Käytettävyyden määritelmä..... | 15 |
| 3.2 Käytettävyyden ilmeneminen | 17 |
| 3.2.1 Ihminen koneena..... | 17 |
| 3.2.2 Ihminen käyttäjänä | 17 |
| 3.3 Käytettävyyssuunnittelu | 19 |
| 3.4 Käyttäjäkeskeinen suunnittelu | 19 |
| 3.5 Käytettävyyden uudet haasteet..... | 20 |
| 4 ANDROID | 22 |
| 4.1 Androidin arkkitehtuuri | 23 |
| 4.2 Android-pohjaiset laitteet | 24 |
| 4.2.1 Puhelimet | 24 |
| 4.2.2 Taulutietokoneet..... | 25 |
| 4.3 Android Market | 26 |
| 5 KÄYTTÖLIITTYMÄN SUUNNITTELU | 27 |
| 5.1 Käyttöliittymäsuunnittelun vaiheet..... | 27 |
| 5.2 Yleisimmät virheet käyttöliittymäsuunnittelussa | 29 |
| 5.3 Käytettävyydestaus | 29 |

| | |
|---|----|
| 6 TYÖSSÄ KÄYTETYT MENETELMÄT JA TYÖKALUT | 32 |
| 6.1 Inflexion UI..... | 32 |
| 6.2 Eclipse | 33 |
| 6.3 i.MX53 Quick Start Board | 33 |
| 7 TYÖN TOTEUTUS | 34 |
| 7.1 Projektin, layoutien ja eventien luonti..... | 34 |
| 7.2 Projektin liittäminen Eclipseen | 36 |
| 7.3 Android-sovelluksen luonti ja liittäminen alustalle | 37 |
| 7.4 Testaus | 37 |
| 8 YHTEENVETO | 39 |
| LÄHTEET | 40 |

LYHENTEET JA TERMISTÖ

| | |
|----------|--|
| ANSI | (American National Standards Institute) Yhdysvaltain kansallinen standardi-instituutti |
| ARM | (Advanced RISC Machines) 32-bittinen mikroprosessoriarkkitehtuuri |
| ASCII | (American Standard Code for Information Interchange) 7-bittinen eli 128 merkkipaikan laajuinen tietokoneiden merkistö, joka sisältää ensisijaisesti amerikanenglannissa tarvittavat kirjaimet, numerot, väli- ja erikoismerkkejä sekä eräitä ohjauskoodeja |
| BIOS | (Basic Input-Output System) Tietokoneohjelma, joka etsii ja lataa käyttöjärjestelmän keskusmuistiin sekä käynnistää sen tietokoneen käynnistyessä |
| CGA | (Color Graphics Adapter) Ensimmäinen väristandardi IBM PC:lle |
| CLI | (Command Line Interface) Komentoliittymä, tapa järjestää ihmisen ja tietokoneen välinen kommunikointi |
| EGA | (Enhanced Graphics Adapter) Näyttöstandardi |
| GHz | Gigahertsi |
| GUI | (Graphical User Interface) Graafinen käyttöliittymä |
| iOS | Applen kehittämä käyttöjärjestelmä |
| iPad | Applen valmistama taulutietokone |
| ISO | (International Organization for Standardization) Kansainvälinen standardisoimisjärjestö |
| Micro-SD | SD-korttiperheen toistaiseksi pienin korttityyppi |
| RISC | (Reduced Instruction Set Computer) Tietokoneiden suoritinarkkitehtuuri |
| RJ-45 | Parikaapeli |
| SD | (Secure Digital) Muistikorttityyppi |
| SUS | (System Usability Scale) Yksinkertainen kymmenkohtainen kaava-ke, jonka avulla saadaan subjektiivinen arvio tuotteen yleisestä käytettävyydestä |
| TUI | (Text User Interface) Tekstipohjainen käyttöliittymä |

| | |
|--------|---|
| UCD | (User-Centered Design) Käyttäjakeskeinen suunnittelu |
| UI | (User Interface) Käyttöliittymä |
| USB | (Universal Serial Bus) Sarjaväyläarkkitehtuuri oheislaitteiden liittämiseksi tietokoneeseen |
| VGA | (Video Graphics Array) Näyttöstandardi |
| Widget | Iso käyttöliittymäelementti, esimerkiksi valintaikkuna |

1 JOHDANTO

Android mullistaa maailmaa. Näin voitaneen sanoa, kun ajatellaan mobiililaitteiden muutaman viime vuoden kehitystä. Android-ohjelmistopinoa hyödynnettiin myös tässä tehdyssä insinööriyössä, kun suunniteltiin graafinen käyttöliittymä.

Tämän opinnäytetyön tavoitteena on kuvata käyttöliittymäsuunnittelun perusasioita ja suunnitella graafinen käyttöliittymä Android-pohjaiseen laitteeseen. Aihevalinnan taustalla oli Navicron Oy:n asiakkaan tekemä tilaus. Navicron Oy on oululainen langattoman teknologian tuotteita kehittävä yritys, jossa olen työskennellyt toukokuusta 2010 alkaen. Käyttöliittymän suunnittelun lisäksi tarkoituksena on yrittää samalla selvittää, sopisivatko käytetyt työkalut Navicronin tulevaisuuden projekteihin.

Työssä käydään läpi aiheeseen liittyvää teoriaa, kuten eri käyttöliittymätyyppejä, ja perehdytään käsitteeseen käytettävyys. Käytettävyyden selvittäminen on paikallaan, koska käyttöliittymiä rakennettaessa on ihmisen eli käyttäjän toiminta avainasemassa. Lisäksi kuvataan Android-käyttöjärjestelmää ja käyttöliittymän suunnittelua yleensä. Teoriaosassa tutustutaan myös suunnittelun aikana käytettyihin menetelmiin sekä työkaluihin.

Työ toteutettiin siten, että teoriaa varten tutkittiin sekä aiheeseen liittyvää kirjallisuutta että Internet-materiaalia. Teoriaosuus on kuvailevaa. Itse suunnittelutyössä käytettiin Mentor Graphicsin Inflexion UI -ohjelmaa. Freescalen i.MX53 Quick Start Boardiin implementoitiin Android-paketti, joka luotiin Eclipsen avulla. Suunnittelutyön tietoperustana toimivat Mentor Graphicsin Inflexion UI -training-sivuston videot sekä muu oppimateriaali.

2 KÄYTTÖLIITTYMÄN PERUSTEET

Laitteissa, ohjelmistoissa sekä muissa samankaltaisissa tuotteissa on osa, jota kutsutaan käyttöliittymäksi. Käyttöliittymän avulla tuotteita käytetään. Kun on kyse matkapuhelimesta, käyttöliittymällä tarkoitetaan sitä osaa, jonka käyttäjä näkee matkapuhelimen ruudulla, sekä sitä tapaa, millä käyttäjä käyttää ohjelmistoa, esimerkiksi matkapuhelimen näppäimistöä tai kosketusnäyttöä. (1.)

2.1 Käyttöliittymän tärkeys

Aiemmin oli vallalla suunnittelunäkemyks, jonka mukaan käyttöliittymä on välttämätön paha, joka häiritsee tietojärjestelmän toteuttamista. Tämän seurauksena järjestelmän käyttö saattaa viedä liikaa aikaa ja huomiota käyttäjältä sekä häiritsee varsinaisen tehtävän suorittamista. Tutkimuksien mukaan melko vähäinen käyttöliittymän muutos voi saada aikaan huomattavia eroja järjestelmän oppimismuutoksissa, järjestelmän käyttönopeudessa, virheiden lukumäärässä ja käyttäjien tyytyväisyydessä. (2.)

Nykyisin käyttöliittymät saavat enemmän huomiota ohjelmistosuunnittelijoilta, koska hyvä käyttöliittymä tarkoittaa enemmän ostajia, säästöä kokonaisrahoituksessa, vähemmän muutoksia, vähemmän kyselyjä ja hyvää mainosta yritykselle. Hyvin suunniteltu käyttöliittymä häivyttää laitteen käyttäjän ja sovelluksen välistä. (2.)

Perinteisesti huonon käyttöliittymän tunnistaa esimerkiksi huonoista ja yhteensopimattomista komentojen nimeämisistä, helposti sekoittuvista komentojonoista, kaoottisista näytöistä, ristiriitaisista komennoista, epätäydellisistä ohjeista, harhaanjohtavista ja pelottavista virheilmoituksista, mitäänsanomattomista virheilmoituksista tai vaikeista virheistä toipumismenetelmistä (2.)

Käyttöliittymiä on ajan saatossa kehitetty erilaisia eri tarpeisiin, kuten komentoliittymä, tekstipohjainen ja graafinen käyttöliittymä, joita seuraavassa esitellään lyhyesti.

2.2 Komentoliittymä

Komentoliittymien (CLI) historia alkaa tietokoneiden reikäkorttiajasta. Tuolloin tietokoneiden käyttö oli suoraviivaista komentojen ja suoritteiden vuorottelua. Komentoliittymien avulla tietokoneita pystyttiin käyttämään ilman reikäkortteja. Vuorovaikutus tietokoneen kanssa nopeutui, kun komennot ja tulosteet voitiin toteuttaa samalla välineellä. Komentoliittymää voi pitää graafisen käyttöliittymän vastakohtana, koska käyttäjä kirjoittaa siinä komentoriville tietyn syntaksin mukaisen komennon, minkä jälkeen komentotulkki käsittelee käskyn ja tulostaa vastineen näytölle. Tämän jälkeen komentoliittymä jää odottamaan uutta komentoa. Komentoliittymien käyttö on helppoa, koska komennot ovat useimmiten englanninkielisten sanojen lyhennyksiä. Komentoliittymiä on edelleen käytössä niiden selkeyden sekä vähäisten resurssivaatimusten takia esimerkiksi pankki-automateissa. Kuvassa 1 on esimerkki komentoliittymästä. (3.)

```
$ cat > wikipedia.pas
program wikipedia;

uses SysUtils;

begin
  WriteLn('Wikipedia is so cool');
end.
cat:
Felipe@mr-69ypraxr6d97 ~
$ fpc wikipedia.pas
Free Pascal Compiler version 2.0.0 [2005/05/08] for i386
Copyright (c) 1993-2005 by Florian Klaempfl
Target OS: Win32 for i386
Compiling wikipedia.pas
Linking wikipedia.exe
7 Lines compiled, 0.3 sec
Felipe@mr-69ypraxr6d97 ~
$ ./wikipedia.exe
Wikipedia is so cool
Felipe@mr-69ypraxr6d97 ~
$
```

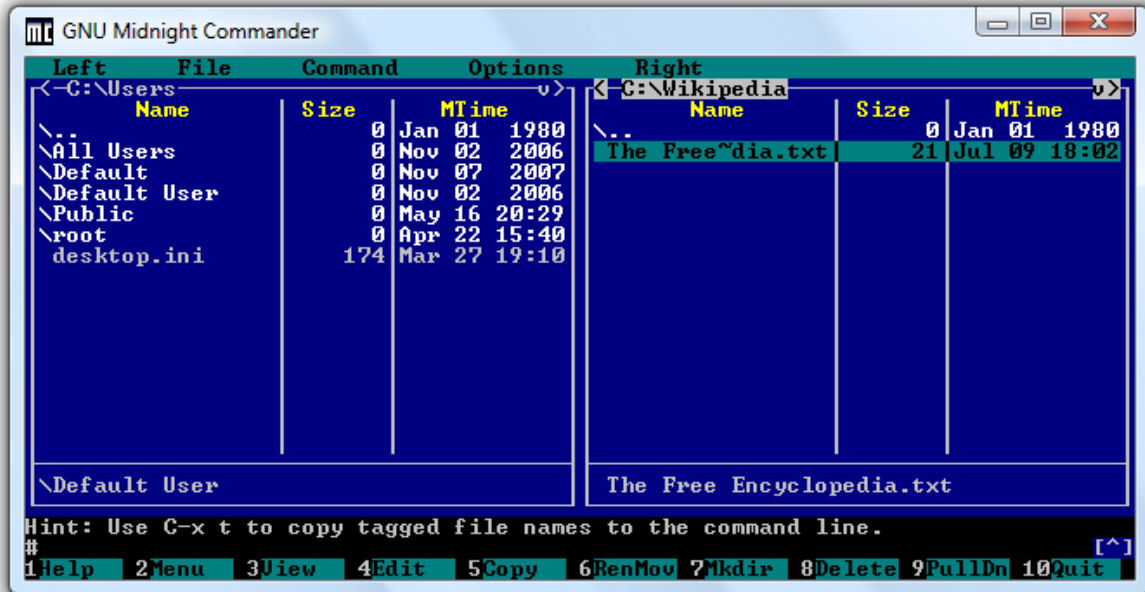
KUVA 1. FPC-komentoliittymä (4)

2.3 Tekstipohjainen käyttöliittymä

Tekstipohjainen käyttöliittymä (TUI) oli siirtymävaiheen käyttöliittymä, kun siirryttiin komentoliittymistä täysin graafisiin käyttöliittymiin. TUI on käyttöliittymä, jossa ei käytetä näytönohjaimien graafisia ominaisuuksia siten, kuin ne nykyisin ymmärretään. Tekstipohjaisessa käyttöliittymässä näytön tulosteet saadaan näytönohjaimen merkkigeneraattorilta, jossa on yleensä ASCII- tai ANSI-standardin mukainen kokoelma merkkejä. Näyttötiloina toimii VGA, EGA tai CGA. (5.)

Tekstipohjaisessa käyttöliittymässä voidaan käyttää kokonäyttötilaa, eikä riviltä tulostus ole välttämätöntä toisin kuin komentoliittymässä. TUI:ssä pystytään myös emuloimaan karkeasti grafiikkaa ja ikkunoita hyvin vähäisellä näyttömuistilla. Hiiriohjauskin on mahdollista. (5.)

Tekstipohjaisen käyttöliittymän tyypillisiä sovelluksia ovat tekstinkäsittelyohjelmat sekä tiedostonhallintaohjelmat. Nykyään tekstipohjaista käyttöliittymää käytetään jonkin verran käyttöjärjestelmien asennusohjelmissa, sisäänkirjautumisikkunoissa ja BIOSin hallinnassa. Kuvassa 2 on esimerkki tekstipohjaisesta käyttöliittymästä. (5.)



KUVA 2. Tekstipohjainen käyttöliittymä GNU Midnight Commander (6)

2.4 Graafinen käyttöliittymä

Graafinen käyttöliittymä (GUI) syntyi Stanford-yliopiston tutkimuskeskuksessa tekstipohjaisten, hiirellä käytettävien on-line-järjestelmien kehityksen yhteydessä. Esimerkiksi Xerox PARC kehitti käyttöliittymää edelleen ja käytti graafista käyttöliittymää Xerox-Alto-tietokoneen pääasiallisena käyttöliittymänä. Edellä mainitun patentoimattoman käyttöliittymän esittely johti siihen, että Apple, Commodore ja Microsoft kopioivat keksinnön itselleen ja alkoivat kehittää omia graafisia käyttöliittymiään. (7.)

Graafisessa käyttöliittymässä teksteillä, kuvilla ja käyttöliittymäelementeillä käytetään tietokoneita ja tietokoneohjattuja laitteita. Käyttöliittymäelementtejä ovat ikkunat, valikot, valintanapit, valintalaatikot ja kuvakkeet. Näitä käytetään näppäimistön lisäksi osoitinlaitteilla, kuten hiirillä, kosketuslevyillä sekä osoitinpaloilla. Nykyaikaisten tietokoneiden käyttöliittymä muodostuu yleensä työpöytäympäristöstä, jonka kautta käytetään graafisia ohjelmia. Kuvassa 3 on esimerkki graafisesta käyttöliittymästä. (7.)



KUVA 3. Windows 7:n graafinen käyttöliittymä (8)

Tämän opinnäytetyön osana olevassa suunnittelutyössä toteutetaan nimenomaan graafinen käyttöliittymä.

2.5 Käyttöliittymät tulevaisuudessa

Voimme vain arvailla, millaisiksi käyttöliittymät tulevaisuudessa kehittyvät. Seuraavan sukupolven käyttöliittymät voisivat esimerkiksi olla 3d-käyttöliittymiä. Tähän mennessä 3d-käyttöliittymiä on nähty lähinnä peleissä, mutta tulevaisuudessa kolmiulotteisia käyttöliittymiä voidaan käyttää esimerkiksi kännyköissä, kodin lämpötilan tai sähkön hallinnassa tai teollisuuden laitteiden etähallinnassa. (9.)

Monet valmistajat kuten Microsoft ovat jo pitkään kaavailleet eleohjauksen sisällyttämistä käyttöliittymiin. Microsoft onkin jo julkaissut Xbox360-pelikonsolille Kinect-lisävarusteen, jossa ohjaus tapahtuu pelaajan vartalolla, jota kuvataan stereokameralla. Näin ollen mitään varsinaista ohjainta ei tarvita. (10.)

3 KÄYTETTÄVYYS

Käyttöliittymiä suunnitellaan ihmistä varten ja siksi ihmisen toimintaa on syytä tuntea. Ihmisen tuntemuksia ja toimintaa voidaan selventää tutkimalla käsitettä käytettävyys.

3.1 Käytettävyyden määritelmä

Käytettävyydelle on useita erilaisia määritelmiä. Määritelmien on vaikea tavoittaa asian ydintä, koska käytettävyys on abstrakti käsite ja sitä on vaikea kytkeä mihinkään konkreettisiin laatuihin. Sen sisältöä voidaan kuitenkin valottaa selvittämällä sen suhteita muihin käsitteisiin. (11, s. 17–18.)

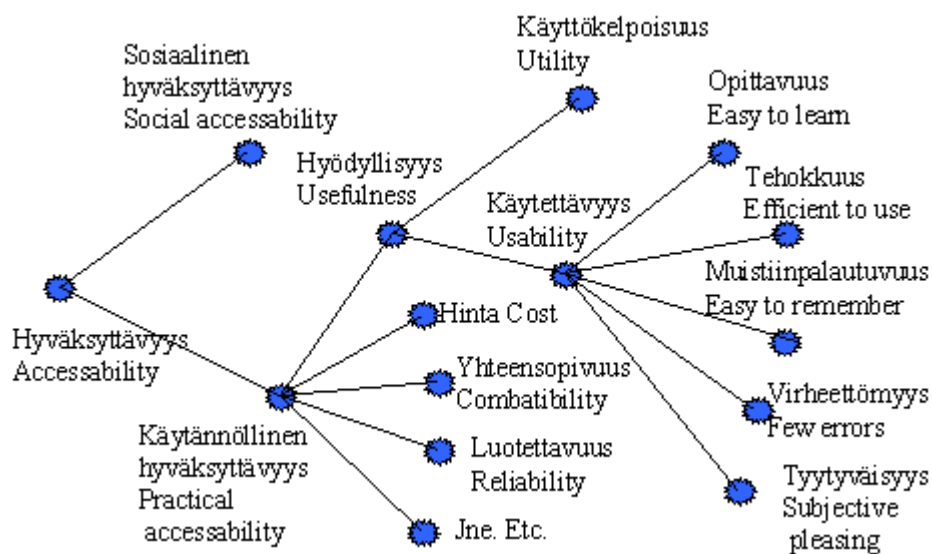
Käytettävyys on menetelmä ja teoriakenttä, jonka kautta käyttäjän ja laitteen yhteistoimintaa pyritään saamaan tehokkaammaksi ja käyttäjän kannalta miellyttävämmäksi. Käytettävyys käyttää hyväksi kognitiivisen psykologian sekä ihmisen ja koneen vuorovaikutuksen tutkimusta. (11, s. 17.)

Sekä puhekielessä että useissa julkaisuissa ihmisen ja koneen vuorovaikutus ja käytettävyys nähdään pitkälti samana asiana. Teoriassa voidaan todeta, että ihmisen ja koneen vuorovaikutus ei ajattele ihmistä organisaation osana, työntekijänä ja tahtovana toimijana. Käytettävyys puolestaan ottaa huomioon nämäkin laitteen ja ihmisen vuorovaikutukseen kuuluvat osat. (11, s. 18.)

Käytettävyys kuvaa, kuinka helppoa ja tehokasta on käyttää tuotetta. Koska käytettävyys on hankala määritellä, se usein määritellään osakokonaisuuksina. ISO 9241-11 -standardin mukaan käytettävyys on mittari, jolla määritellään tuotteen käytön tuottavuutta, tehokkuutta sekä miellyttävyyttä. ISO 9241-11 -standardissa tuottavuus tarkoittaa sitä, että tehtävät tulevat tehdyksi täydellisesti ja virheettömästi. Tehokkuus kertoo, kuinka paljon resursseja tuotteessa tarvitaan henkilönä, rahana sekä aikana. Miellyttävyyden puolestaan kertoo, kuinka miellyttävä tuotetta on käyttäjän mielestä käyttää. (12.)

Käytettävyydelle ei siis ole yleisesti hyväksyttyä määritelmää, vaan määrittely riippuu määrittelijästä. Tietotekniikan alalla käytettävyyden tutkimisen tunnetuin uranuurtaja on Jakob Nielsen, joka on laajentanut ISO 9241-11 -standardia. Nielsen määrittelee käytettävyyden hyväksyttävyyden käsitteen avulla. Kuva 4 esittää Nielsenin mallia hyväksyttävyyden eri osatekijöistä. (13.)

Käsitteen hyödyllisyys Nielsen jakaa kahteen kategoriaan, käyttökelpoisuuteen ja käytettävyyteen. Käyttökelpoisuus kuvaa, suoriutuuko järjestelmä niin kuin pitäisi ja käytettävyys puolestaan kuinka hyvin käyttäjät voivat käyttää toimintoa. (14, s. 24–25.)



KUVA 4. Malli hyväksyttävyyden osatekijöistä Nielsenin mukaan (15)

Itse käytettävyydelle Nielsen antaa viisi laatukomponenttia:

- Opittavuus: Systemin pitää olla helppo oppia, jotta käyttäjä voi nopeasti alkaa käyttää sitä tulosten aikaansaamiseksi.
- Tehokkuus: Opittuaan käyttämään järjestelmää käyttäjä saa aikaan tuloksia tehokkaasti.
- Muistettavuus: Järjestelmä on helppo muistaa, vaikka välillä on käyttämättä sitä.
- Virheettömyys: Käyttäjät tekevät vähän virheitä käyttäessään systeemiä. Jos virheitä tehdään, niistä toivutaan helposti.
- Tyytyväisyys: Järjestelmää on miellyttävä käyttää, niin että käyttäjät ovat tyytyväisiä käyttäessään sitä. (14, s. 26.)

3.2 Käytettävyyden ilmeneminen

Uusi laite tai järjestelmä pitää rakentaa mahdollisimman ”luonnolliseksi”. Jos huono ominaisuus todetaan vasta käyttöönoton tai markkinoinnin aloittamisen jälkeen, aiheutuu siitä paljon kustannuksia. Jo suunnittelun alkuvaiheessa huono tai toiseen vaihdettu ominaisuus ei ehdi tulla kalliiksi, ei vie suunnittelun tai toteuttamisen resursseja eikä aiheuttaa turhautumista suunnittelijoissa (11, s.20.)

3.2.1 Ihminen koneena

Ihmisestä koneena, joka prosessoi ja varastoi saamaansa dataa sekä toimii sen pohjalta, muodostui yleinen malli sen jälkeen, kun tietokone esiteltiin tuotantokäyttöön. Tämä malli pystyy osittain kuvaamaan ihmisen muistin joidenkin osalueiden toimintaa nykyisin ymmärrettävällä tavalla. (11, s. 21.)

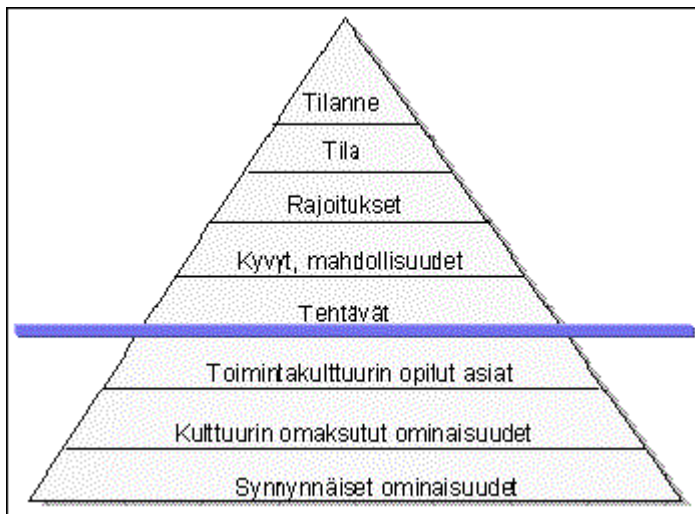
Ihminen on kuitenkin ihminen, elävä, tunteva ja kokeva olento. Semiootikot sanovat ihmistä pakonomaiseksi merkitysten muodostajaksi, joka ei voi olla sitomatta merkityksiä havaitsemiinsa asioihin ja ilmiöihin. (11, s. 21.)

Tekniikan kehittyessä ihmisen ja koneen ero on tullut yhä vaikeammaksi määrittellä. Ongelmakohtaksi on muodostunut ihmisen määrittely tässä vertailussa. Tekniikka voi olla osa ihmistä itseään, kuten sydämentahdistin, sisäkorvaimplantti ja tekonivel. (16.)

3.2.2 Ihminen käyttäjänä

Ihmisen toiminnasta tiedetään joitakin asioita, jotka näyttävät pätevän enemmän tai vähemmän kaikkiin ihmisiin. Ihmisiä on kuitenkin maailmassa yli seitsemän miljardia, ja melkein kaikki säännöt tuntuvat tuntevan poikkeuksen.

Epävarmuus korkeissa paikoissa ja ”huimaus” ovat yleisesti tunnettuja ihmisen ominaisuuksia. Punaisen ja vihreän tai valkoisen ja mustan symbolisia arvoja pidetään länsimaissa hyvin pitkälti sovittuina. Itämaissa merkitykset tuntuvat kuitenkin kääntyvän länsimaisesta näkökulmasta katsottuna pääläelleen. Kuvassa 5 kuvataan ihmistä käyttäjänä. (11, s. 22.)



KUVA 5. Ihminen käyttäjänä (12)

Käytettävyyttä suunniteltaessa huomioon otettavaa on se, että käyttäjänä on ihminen, koska ihmisten henkilökohtaiset ominaisuudet vaihtelevat suuresti. Käytettävyyden suunnittelussa pitää pyrkiä segmentoimaan käyttäjäryhmä riittävän selkeästi, jotta voidaan käyttää tiettyjä ominaisuuksien stereotyyppioita pohjana toiminnallisuuksien suunnittelussa. Huomioon otettavia asioita, joita käyttäjä-ihminen kantaa mukanaan, ovat esimerkiksi

- synnynnäiset fysiologiset sekä psykologiset rakenteet, esimerkiksi aistit, muistirakenteet ja perustarpeet
- suhteellisen pysyvät kulttuuriset asiat, esimerkiksi kieli, sekä osa normeista ja tavoista
- erilaisiin teknisiin toimintaympäristöihin liittyvät konventiot. Kyseisiä toimintatapoja voidaan vahvistaa, kun suunnittelijat noudattavat niitä.

Ihmisen toimintaan vaikuttavat myös monet muutkin asiat, kuten

- vaihtelevat kulttuuri- ja ympäristöelementit (muoti, alakulttuurit sekä talokohtaiset toimintatavat)
- tehtävät
- yksilölliset toimintarajoitukset ja -kyvyt
- tila, jossa toimitaan, ja sen olosuhteet
- käyttötilanne. (11, s. 23–24.)

3.3 Käytettävyyssuunnittelu

Käytettävyyssuunnittelulla tarkoitetaan niitä aktiviteetteja, periaatteita ja menetelmiä, jotka tulisi suorittaa osana tuote- tai järjestelmäkehitystä hyvän käytettävyyden varmistamiseksi. Käytettävyyssuunnittelussa pitää muistaa, että käytettävyys on käyttäjäystävällisyyttä ja hyvää käyttökokemusta. Suunnittelussa on tärkeää

- tuntee tuleva käyttäjä ja ihmisen kognitiiviset kyvyt
- pitää huoli, että käyttäjän tarkkavaisuus ei ohjaudu "väärin"
- ottaa huomioon, ettei suunnitellun tuotteen jokin ominaisuus sulje pois joitakin käyttäjäryhmiä (ellei tuotetta ole suunniteltu erityisesti esimerkiksi oikea- ja vasenkätisille, on kummankin ryhmän voitava käyttää tuotetta vaivattomasti)
- muistaa mihin tuote on tarkoitettu
- analysoida mahdollisesti jo tehtyjen käyttäjätutkimusten tuloksia
- muistaa asiayhteyden tärkeys käytettävyydelle
- tuntee kulttuuriset lukutavat. (17, s.13–20.)

Käytettävyyssuunnittelumenetelmiä on useita. Esimerkiksi käytettävyystestaus selvittää, kuinka käyttäjät selviytyvät testattavan tuotteen tai palvelun käyttämisestä. Testitilanteessa käyttäjä suorittaa hänelle annettuja tehtäviä ja samanaikaisesti hänen tekemisiään tarkkaillaan. Käytettävyyden vaatimusmäärittelyssä puolestaan on tavoitteena alustavasti määrittää käytettävyystavoitteet kehitettävälle tuotteelle. Tällöin käytetään esimerkiksi workshop ryhmätyöskentelyä (post-it-laput, seinätaulutekniikka ja priorisointi). Yksi tunnetuimmista käytettävyyden arviointimenetelmistä on heuristinen arviointi, joka perustuu Nielsenin heuristiikkaan. Tällöin käydään läpi, noudattavatko käyttöliittymän osat käytettävyyden periaatteita. Edellisten lisäksi käytetään myös SUS-kyselyä, joka on yksinkertainen kymmenenkohtainen kaavake, jonka avulla saadaan subjektiivinen arvio tuotteen yleisestä käytettävyydestä. (18, s. 9–12.)

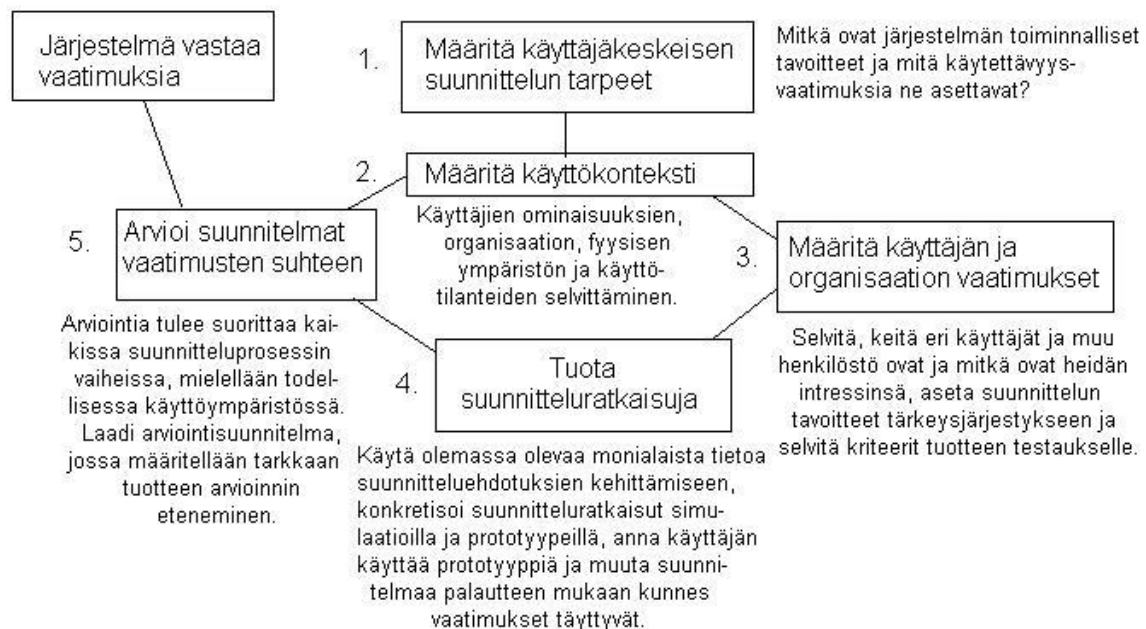
3.4 Käyttäjäkeskeinen suunnittelu

Käyttäjien toiveet ja tarpeet ovat käyttäjäkeskeisen suunnittelun (UCD) perusta. Suunnitteluprosessissa tämä käyttäjänäkökulma on mukana siten, että käyttäjä

osallistuu koko ajan. Arviointi on osa suunnittelua eikä erillinen tapahtuma. Suunnittelun aikana arviointi on jatkuvaa sekä suunnittelu toistuvaa. Tarkoituksena on taata tuotteen hyödyllisyys ja helppokäyttöisyys. (19.)

ISO 13407 -standardi (kuva 6) määrittelee käyttäjäkeskeisen suunnitteluprosessin neljänä tarvittaessa uudelleen toistettavana vaiheena:

- käyttökontekstin ymmärtäminen ja määrittely
- käyttäjävaatimusten ja organisaation vaatimusten määrittely
- suunnitteluratkaisujen luominen
- evaluointi (20.).



KUVA 6. ISO 13407 -standardi (20)

3.5 Käytettävyyden uudet haasteet

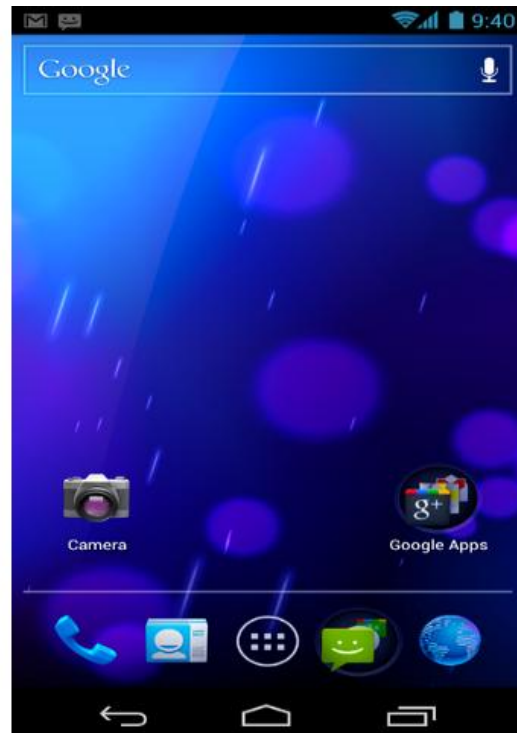
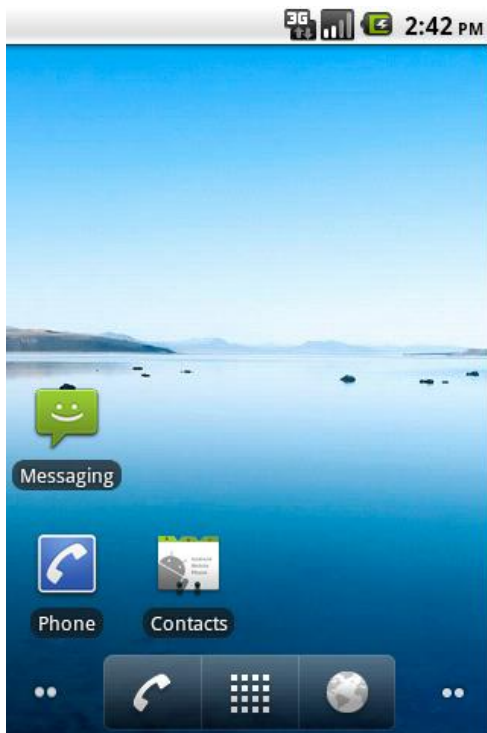
Käytettävyydellä on nyky-yhteiskunnassa paljon uusia haasteita, koska tekniikka hallitsee niin paljon ihmisten arkipäiväistä elämää, ja se on lähes kaikkialla läsnä. Yksi näistä haasteista on niin sanottu älykäs ympäristö, joka alkaa jo arkipäiväistyä. Älykkäiden ympäristöjen toiminnot voidaan jakaa kuuteen perustyyppiin niiden käyttötarkoituksen perusteella: ohjaus, kommunikointi, informaation kysely, ilmoituspalvelu, informaation kerääminen ja automaatio. Haasteena on myös se, että laitteita voidaan hallita näkyvinä ja näkymättöminä. Ihmiset

ovat aktiivisia ja aloitteellisia toiminnoissaan. He ottavat vastuun käyttäytymisestään, ja näin tehdyt valinnat ovat omien arvojen mukaisia eikä olosuhteiden tai muiden mielipiteiden sanelemia. Tämä ihmisen proaktiivisuus on lisähaaste käytettävyydelle. Lisäksi tilanteiden ja paikan hallinta sekä yksityisyyskysymykset ja tietosuoja on otettava huomioon käytettävyyden suunnittelussa. (17, s. 21.)

4 ANDROID

Android on 5. marraskuuta 2007 julkaistu ohjelmistopino matkapuhelimille sekä muille mobiililaitteille. Se sisältää myös käyttöjärjestelmän, väliohjelmistoja ja käyttäjän perusohjelmistoja. Androidin suosion kehitys älypuhelimissa on ollut huimaa, ja se nousikin älypuhelimien markkinajohtajaksi vuoden 2010 viimeisellä neljänneksellä. Androidin suosio perustuu sen helppokäyttöisyyteen sekä muokattavuuteen. Muokattavuuden ansiosta käyttäjä voi esimerkiksi vaikuttaa puhelimen näytön ulkoasuun sekä puhelimen ominaisuuksiin, joten käyttäjä voi tehdä käytännössä puhelimen, jollaista ei ole kukaan muulla. Toinen suosioon vaikuttava seikka on Android Market, josta käyttäjä voi ladata erilaisia miniohjelmia käyttöönsä. Suurin osa Android Marketin ohjelmista on ilmaisia tai ne maksavat vain hyvin vähän, muutamasta sentistä muutamaan euroon. (21.)

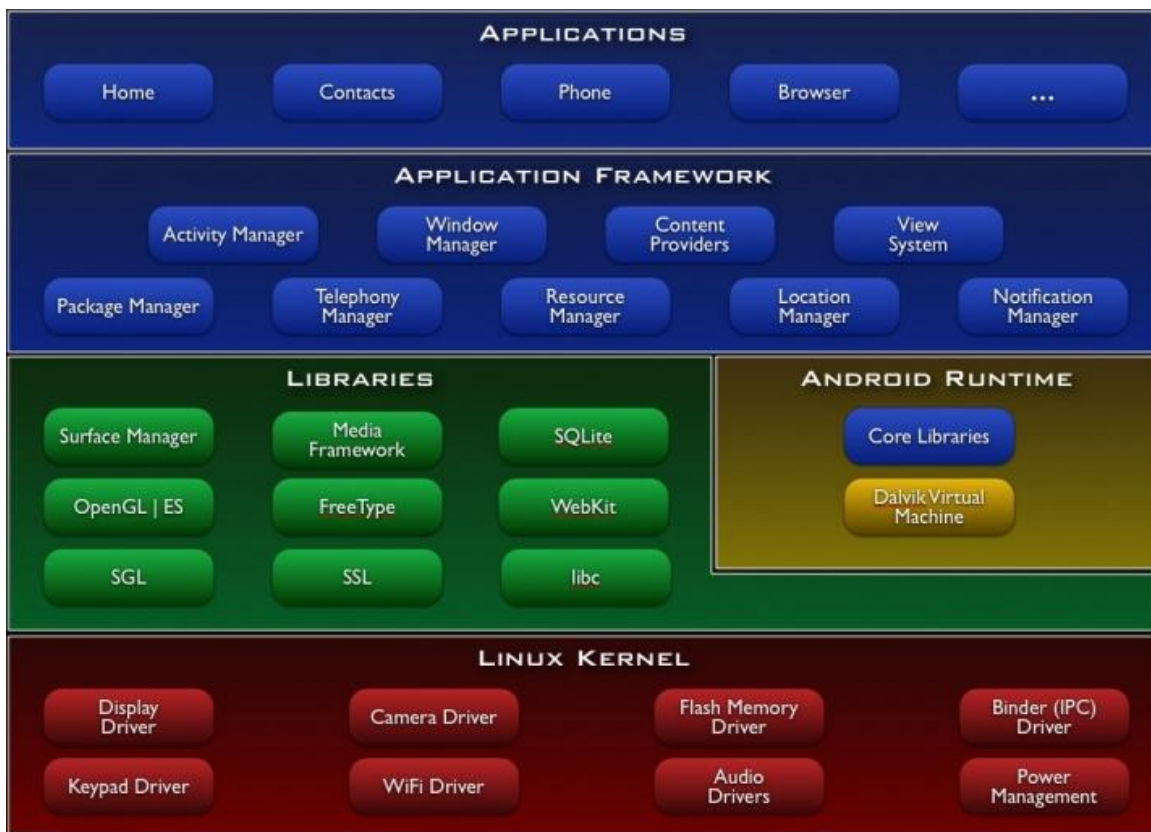
Älypuhelinvalmistajien ja sovelluskehittäjien suosion Android on saanut, koska se on avoin ja lisenssimaksuton vaihtoehto. Androidista julkaistaan kahdesta neljään kertaa vuodessa uusi päivitetty versio, joissa käyttöjärjestelmää on paranneltu ja lisätty uusia tukia. Esimerkiksi Android 2.2 -versioon oli lisätty kauan kaivattu tuki Adobe Flash 10.1:lle (kuva 7). Viimeisin julkaistu versio on Androidin 4.0 (kuva 7), kutsumanimeltään Ice Cream Sandwich. (21.)



KUVA 7. Vasemmalla Android 2.2:n perusnäky (22) ja oikealla Android 4.0 Google Nexus -puhelimessa (23)

4.1 Androidin arkkitehtuuri

Android-arkkitehtuuri (kuva 8) muodostuu viidestä tasosta. Arkkitehtuurin ylimpänä tasona ovat valmiit sovellukset (Applications) kuten kalenteri, kartat ja seläin. Tälle tasolle asettuvat myös sovelluskehittäjien luomat sovellukset. Seuraavana arkkitehtuurissa on sovelluskehystaso (Application Framework), joka tarjoaa sovelluskehittäjien käyttöön valmiita ohjelmistokomponentteja sekä rajapintoja. Androidissa kirjastot (Libraries) ovat C- ja C++ -kirjastoja, joihin päästään käsiksi sovelluskehystasolta. Android Runtime -tasolla tapahtuu sovellusten varsinainen suoritus, ja siellä sijaitsevat myös tärkeimmät kirjastot (Core Libraries). Androidin jokaista sovellusta suoritetaan omana prosessina virtuaalikoneen (Dalvik Virtual Machine) avulla. Virtuaalikone on suunniteltu siten, että sen avulla kyetään suorittamaan useita prosesseja samanaikaisesti. Linuxydintäsoa (Linux Kernel) käytetään tärkeimpiin järjestelmän toimintoihin, kuten prosessienhallintaan ja turvallisuuteen. (24.)



KUVA 8. Androidin arkkitehtuuri (25)

4.2 Android-pohjaiset laitteet

4.2.1 Puhelimet

Android-puhelimia (älypuhelimia) kehittävät useat suuret valmistajat, kuten HTC, LG, Samsung, Motorola ja Sony-Ericsson, sekä joukko pienempiä valmistajia. Android-puhelinta käyttääkseen käyttäjä tarvitsee käyttäjätilin Google-palveluihin, mikäli haluaa käyttää puhelimen suomia mahdollisuuksia hyödykseen. Ilman Googlen käyttäjätiliä Android-puhelin toimii lähinnä peruspuhelimena ja käyttäjä voi hyödyntää vain puhelimen perusominaisuuksia, siis soittaa ja lähettää tekstiviestejä. (26.)

Google-tilin luonnin jälkeen käyttäjälle avautuu täysin uusi maailma, jossa hän voi käyttää puhelimensa ominaisuuksia täysin hyödykseen. Tällöin käyttäjä voi

selata nettiä, osallistua sosiaaliseen mediaan sekä käyttää Android Marketia sovelluksien hankintaan. (26.)

Valmistajasta riippumatta Android-älypuhelimissa on käytännössä sama käyttöjärjestelmä, johon suurimmat valmistajat ovat voineet lisätä jo valmiiksi joitain omia sovelluksiaan. Kaikki Android-puhelimet ovat pohjimmiltaan älypuhelimia ja niitä voidaan käyttää netin selaamiseen, sosiaaliseen mediaan, kuvien ja videoiden katseluun sekä musiikin soittamiseen. (26.)

Android-puhelimien hintahaitari on laaja. Halvimmat maksavat noin 100 euroa ja kalleimmat 600 euroa. Merkittävin ero on suorittimen laskentatehossa, jonka vuoksi halvemmissä malleissa multimediaominaisuudet ovat rajoittuneemmat kuin kalleimmissa. Myös näytöissä on eroja, kuten niiden koossa ja tarkkuudessa. Halvemmissä malleissa materiaalit ovat heikompiä, esimerkiksi muovikuoret ja muovinen näytön lasi. Kalleimmissä malleissa yleensä myös kamera on laadukkaampi samoin muutkin ominaisuudet. (26.)

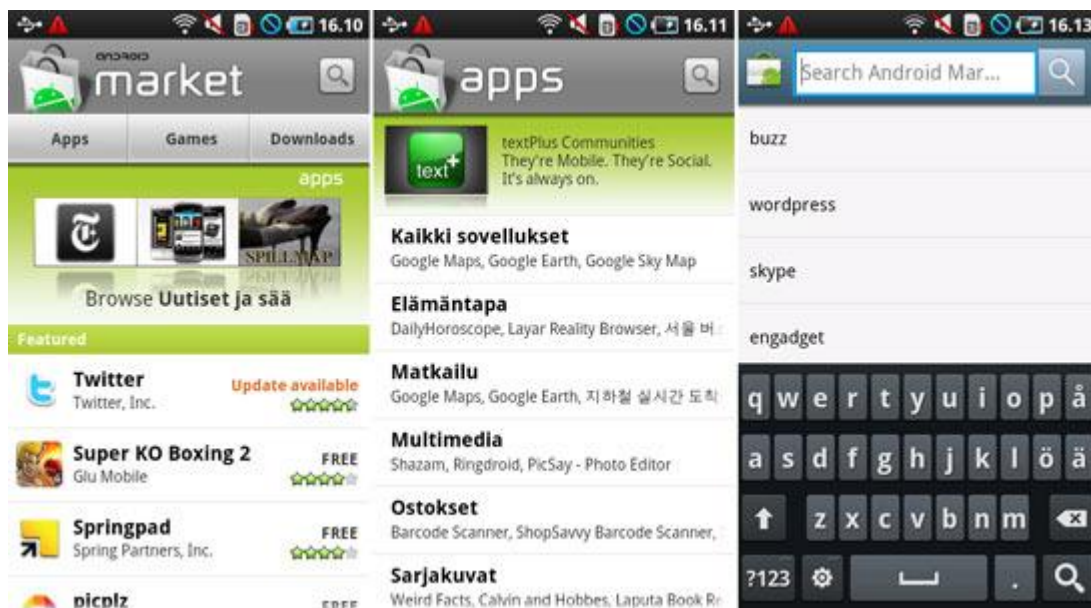
4.2.2 Taulutietokoneet

Taulutietokoneet eli tabletit ovat tekniikaltaan puoleksi tietokoneita ja puoleksi nykyaikaisia älypuhelimia. Taulutietokoneita on ollut olemassa jo usean vuoden ajan, mutta vasta Applen iPadin myötä niiden suosio lähti kasvuun. Tällöin myös useat muut valmistajat lähtivät kehittämään omia Android-pohjaisia versioita tableteista iPadin iOS-pohjaisen kilpailijoiksi.

Taulutietokoneet ovat kosketusnäytöllisiä 7–10":n kokoisia kannettavia tietokoneita. Nykyisin tableteita on kehitetty myös siten, että niihin saa halutessaan irrotettavan näppäimistön. Nämä niin sanotut transformer-tabletit muistuttavat minikannettavia, joissa on kosketusnäyttö.

4.3 Android Market

Android Market on Googlen kehittämä verkossa oleva sovelluskauppa, joka on tarkoitettu Android-käyttöjärjestelmän mobiililaitteille. Laitevalmistajat ovat yleensä esiasentaneet Android Market -sovelluksen valmistamiinsa tuotteisiin. Käyttäjät pystyvät sen avulla lataamaan ja asentamaan uusia sovelluksia Android-laitteisiinsa. Android Marketin sovellusten tarjonta on varsin laaja. Sovelluksia on ollut tarjolla jo yli 500 000 eri aihealueilta, esimerkiksi pelit, urheilu, sää ja widgetit. Tällä hetkellä ladattavissa on noin 350 000 sovellusta. Android Market sai kymmenen miljardia latausta täyteen 6. joulukuuta 2011. Kuvassa 9 on esimerkki Android Marketin mobiilisovellusnäkökymästä Samsung Galaxy S -puhelimessa. (27.)



KUVA 9. Android Market -mobiilisovelluksen näkymät Samsung Galaxy S -puhelimessa (28)

5 KÄYTTÖLIITTYMÄN SUUNNITTELU

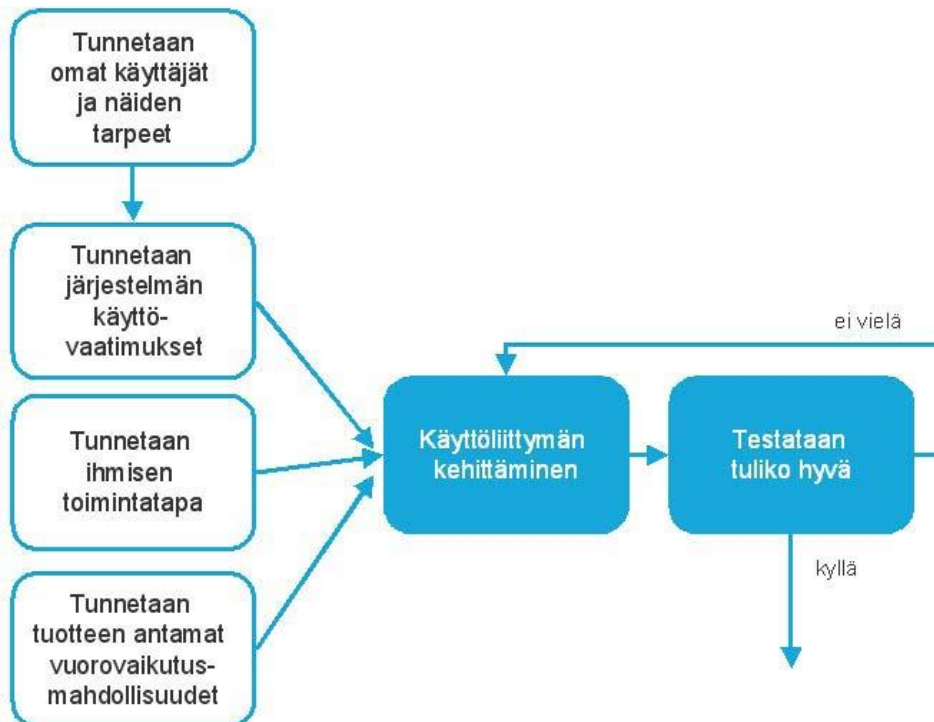
5.1 Käyttöliittymäsuunnittelun vaiheet

Käyttöliittymässä on kyse toiminnasta, ei tekniikasta. Käyttöliittymän suunnittelu on haastavaa, koska siinä yhdistyy kolme hyvin erilaista tieteenalaa: tietotekniikka, psykologia ja kuvaamataide. Työn selkärangan muodostaa vuorovaikutussuunnittelu. Tuotteen logiikan luovat interaktiosuunnittelijat. Tällöin tuotteen toiminnot strukturoidaan siten, että ne on helppo oppia ja muistaa. Graafiset suunnittelijat visualisoivat tuotteen käyttöliittymän sisältämän tiedon ja näin pikselit heräävät erilaisten kuvien välityksellä eloon. Kansainvälistämisessä eli internationalisoinnissa muokataan teksti sellaiseen muotoon, että se on helppo ymmärtää kielestä riippumatta. Käyttöliittymän komennot pitää vielä kääntää ja sopeuttaa eri kielille lokalisoinnin avulla. Suunnitteluryhmään kuuluu myös usein asiantuntijoita, kuten sosiologeja, antropologeja ja kaupallisen alan ihmisiä, joiden työnä on asiakkaiden tarpeiden ymmärtäminen ja huomioon ottaminen tuotevaatimuksissa. Ohjelmoijien on syytä olla mukana suunnitteluryhmässä, jotta tuotteen tekninen toteutus onnistuu. (29, s. 172.)

Käyttöliittymän suunnittelussa on otettava huomioon useita ominaisuuksia, joista käytännöllisyys ja visuaalisuus ovat merkittävimpiä. Suunnittelu on kolmivaiheinen:

- käyttöliittymän käyttötarkoituksen määrittäminen (opetusohjelma, viihde, mainos tai jokin muu)
 - mahdollisten ongelmakohtien (ihmisten kognitiiviset kyvyt, tekninen toteutus) määrittäminen, tarvittaessa niiden jakaminen pienempiin osakokonaisuuksiin ja ongelmien ratkaisumahdollisuuksien selvittäminen
 - kohderyhmien kartoittaminen (lapset, ikäihmiset, noviisit tai asiantuntijat).
- (30.)

Kuva 10 käyttöliittymän kehittämisestä esittää käyttöliittymän suunnitteluun vaikuttavia tekijöitä.



KUVA 10. Käyttöliittymän kehittäminen (31, s.11.)

Tärkeää on selvittää syntyvät kustannukset ja käytettävissä olevat varat. Päätehtään suunnittelu- ja toteutusaikataulusta sekä arvioidaan tuotteen todennäköinen käyttöikä ja selvitetään käytönaikainen ylläpito ja päivittäminen. Selvitetään, millaisella laitteistolla ja ohjelmistolla pystytään parhaiten toteuttamaan asetetut vaatimukset. (30.)

Testaus on oleellinen osa käyttöliittymän suunnittelua ja visuaalista toteutusta. Käyttöliittymän testauksen tarkoituksena on paljastaa huonot vuorovaikutusmekanismit ja laiminlyönnit sekä ristiriitaisuudet ja epäselvyydet, jotka löytyvät käyttöliittymästä. Aluksi laaditaan testaussuunnitelma, jota apuna käyttäen luodaan tarvittavat testitapaukset. Testauksella pyritään löytämään mahdolliset ongelmakohdat ja virheet. Esimerkiksi käyttöliittymä voi toimia, mutta sen toiminnallisuudet eivät ole loogisia. Ongelmakohtien löytämisen jälkeen suunnitellaan parannusehdotukset. (32.)

5.2 Yleisimmät virheet käyttöliittymäsuunnittelussa

Käyttöliittymässä saattaa syntyä myös virheitä. Alla on lueteltu niistä yleisimmät.

- Käyttäjä ei useinkaan ole ATK-alan ammattilainen, toisin kuin suunnittelija. Tällöin tuote helposti suunnitellaan liian teknisestä perspektiivistä, eikä käyttäjä innostu laitteesta, koska sitä on liian monimutkaista käyttää.
- Jos tuote tuodaan liian aikaisin markkinoille vaikkapa kilpailun vuoksi, laite voi olla tavallaan keskeneräinen.
- Ei ole tiedostettu kulttuurieroja.
- Ei ole otettu huomioon eri mitta-, numero-, aika-, kirjainjärjestelmää.
- Eri kulttuureissa suhtaudutaan eri lailla tittleihin ja joihinkin sanoihin.
- Eri-ikäiset ihmiset suosivat erilaisia käyttöliittymiä.
- Miesten ja naisten väliset erot ovat jääneet huomioimatta.
- Eri luonnetyypit voivat suosia erilaisia käyttöliittymiä.
- Koulutustaso, mielentila sekä motivaatio vaikuttavat käyttöliittymän valintaan.
- Käyttöliittymää on voitava käyttää mahdollisimman luontevasti.
- Ihmisten luonnolliset kommunikointitavat ovat puhe ja ilmeet, tekstin käyttö on tullut vasta teollistumisen myötä.
- Laitteiden esitysmuoto on yleensä kaksiulotteinen, vaikka ihminen hahmottaa yleensä kolmiulotteisesti.
- Voi syntyä eturistiriitoja esimerkiksi tietoturvasta. (33, s. 7–10.)

5.3 Käytettävyydestä

Käytettävyydestä varmistetaan sovelluksen helppokäyttöisyydestä, oikean muotoisesta sisällön esittämisestä sekä viestien, kehoitteiden ja kommentojen selyydestä. Tuotteen käytettävyyden rakentamiseksi ja varmistamiseksi on olemassa suuri joukko menetelmiä. Tavallisesti käytetään käytettävyydestin perusversiota, jolla pyritään selvittämään käyttäjän mentaalimalleja siten, että käyttäjä ajattelee ääneen samalla kun suorittaa tehtäviä. On myös toisia tapoja

selvittää mentaalimalleja, esimerkiksi piirtäminen. Ääneenajattelun on havaittu tutkimuksissa antavan parasta tietoa malleista. Mentaalimalliin saattaa herkästi vaikuttaa itse testausprosessi, varsinkin jos tuote on käyttäjälleen uusi. Käytettävyyden testaaminen on joustava menetelmä, jota muuntelemalla saadaan tuotteen käytöstä monenlaista tietoa. (11, s. 275–280.)

Käytettävyydestestauksen nimellä kulkee Suomessa useita käytettävyyssmenetelmiä. Läheskään kaikki niistä menetelmistä eivät ole varsinaisia käytettävyyss-testejä vaan käyttäjien mielipiteen mittauksia esimerkiksi mielipidekyselyin. Usein sekoitetaan toisiinsa tuotteen käytettävyys ja käyttäjän mielipide tuotteesta. Saatetaan uskoa, että käyttäjän mielipiteet tuotteesta kertovat sen käytettävyydestä. Tuotteen miellyttävyys on osa käytettävyyttä, mutta vain osa. Kun ihmisten mielipiteet samasta tuotteesta eroavat toisistaan, käytettävyyss-testissä tapahtuu juuri päinvastoin. Testissä näkyvät jo toisella ja kolmannella peräkkäisellä testikäyttäjällä tuotteen ongelmat, jotka kasaantuvat samoihin käyttöliittymän piirteisiin. (11, s. 275–280.)

Käytettävyyss-testillä on tarkoituksena jäljitellä aitoa tilannetta ja tehdä saatuja mentaalimalleja hyödyntämällä tuotteen käyttölaadusta parempi. Käytännössä käytettävyyss-testejä on kahden tyyppisiä, ensinnäkin sellaisia, jotka ovat osa kehitystyötä, ja toisekseen sellaisia, joilla mitataan, ovatko tuotteet valmiita levitykseen. (11, s. 275–280.)

Voidaan testata joko koko tuotetta, sen prototyyppiä tai osaa, kuten keskeisiä toimintoja, vaikeimmiksi uskottuja tai havaittuja toimintoja tai valikkorakenteita. Tietojärjestelmissä testattaessa voidaan käydä läpi järjestelmän käyttöönotto, käsikirjat ja opasteet, ongelmista toipuminen ja kuvakkeet. (11, s. 275–280.)

Käytettävyyss-testejä olisi syytä tehdä koko järjestelmä- ja tuotekehitysprosessin aikana. Syytä on myös testata valmis järjestelmäkokonaisuus ennen julkaisua. (11, s. 275-280.)

Käytettävyyss-testien tulosten esittäminen on haastava tehtävä. Testeissään etsitään pääasiassa vikoja tuotesuunnittelusta. Kuunnellessaan testi-

tuloksia suunnittelijoiden reaktiot vaihtelevat innostuneesta ärtyneeseen. Kukapa haluaisi kuulla omaa työtä haukuttavan. Testitulosten esittäminen kannattaakin tehdä varovasti. Testien luonteen pitäisi olla sellainen, että suunnittelija itse voi ottaa testien avulla parannetusta käyttöliittymästä kaiken kunnian itselleen. (34, s.182.)

Käytettävyydestit ovat ainoa luotettava mittaustapa, jolla tuotteen käytettävyyttä voidaan todentaa. Testeihin käytetty raha palautuu takaisin, jos testit on tehty huolellisesti ja löydetyt käytettävyysongelmat korjataan. Mitä aikaisemmin ongelmat löydetään, sitä halvemmaksi niiden korjaaminen tulee. (11, s. 275–280.)

Käyttölaadultaan paremmat tuotteet ovat testauksen välitön hyöty. Välillisesti arviointi parantaa suunnittelijoiden valmiuksia ottaa käyttäjät huomioon. Testit voivat vaikuttaa myös suunnittelijoiden asenteisiin käyttäjiä kohtaan ja käytettävyyden tärkeys korostuu. (11, s. 275–280.)

6 TYÖSSÄ KÄYTETYT MENETELMÄT JA TYÖKALUT

6.1 Inflexion UI

Inflexion UI on Mentor Graphicsin suunnittelema ja valmistama ohjelma, jolla pystytään tekemään graafisia ohjelmia ilman varsinaista ohjelmointia. Tarvittaessa voidaan myös muuttaa ohjelman koodia. Näin ollen ohjelman käyttäjä pystyy keskittymään sovelluksen logiikkaan ja toteutukseen. Inflexion UI on ihan teellinen kehittäjille, jotka haluavat luoda Android-pohjaisen laitteen lääketieteen, teollisuuden tai kuluttajan tarpeisiin. Mentor Graphicsin ja Freescalen yhteistyön tuloksena Inflexion UI:lla on vaivatonta tehdä käyttöliittymiä, jotka toimivat Freescalen i.MX51- sekä i.MX53-alustoilla. (35; 36, s.1–2.)

Inflexion UI helpottaa grafiikkapainotteisesta suunnittelusta johtuvien toteuttamisongelmien ratkaisemista. Sen avulla voidaan tehdä muista erottuvia käyttöliittymiä. Inflexion UI on nopeasti uudelleen mallinnettavissa uusille markkinoille, käyttäjäryhmille tai jopa uusille tuotteille. Sillä kyetään luomaan pitkälle kehittyneitä tuotteita. Myös suunnitteluun käytettävää aikaa säästyy, kun ei tarvitse erikseen ohjelmoida tarvittavia efektejä, tilasiirtymiä eikä animaatioita. Inflexion UI kykenee hyödyntämään laitteistoa riippumatta sen tehokkuudesta ja toimii siltana suunnittelun ja toteutuksen välillä. (37.)

Inflexion UI:tä käytettiin tässä projektissa, kun luotiin graafinen käyttöliittymä, joka toimii Freescalen i.MX53-alustalla.

6.2 Eclipse

Eclipse oli alunperin IBM:n kehittämä ja myöhemmin avoimen lähdekoodin lisenssillä kehitettävä ohjelmointiympäristö. Nykyisin kehityksestä vastaa Eclipse-organisaatio, jonka jäseniä ovat mm. IBM, Borland, Oracle ja HP. Kehittäjät pääsevät vapaasti muokkaamaan sitä. Rakenteeltaan Eclipse on modulaarinen, hyvin joustava alusta, jonka saa taipumaan moniin tehtäviin. (38; 39.)

Tässä opinnäytetyöprojektissä Eclipseä käytettiin Inflexion UI:lla toteutetun käyttöliittymän Android-paketin luomisessa.

6.3 i.MX53 Quick Start Board

i.MX53 on alusta, jolla kyetään ajamaan avoimen lähdekoodin sovelluksia. Alustaan on integroitu ARM® Cortex™-A8 1 GHz:n prosessori sekä Freescale MC34708 -virranhallintajärjestelmä. Alusta kykenee myös 1020p:n videon dekodaukseen ja 720p:n enkoodaukseen. i.MX53 sisältää myös tyypillisimmät liitännät, kuten kuuloke, mikrofoni, RJ-45, USB sekä SD- ja Micro-SD-korttipaikat. (Kuva 11.). (40.)



KUVA 11. Freescale i.MX53-alusta (41)

7 TYÖN TOTEUTUS

Tämä luku kuvaa graafisen käyttöliittymän varsinaista suunnittelua Android-pohjaiseen laitteeseen. Esitys on toteutettu yleisellä tasolla, tarkka osuus on vain tilaajan käytössä.

Käyttöliittymän toteuttamiseen valittiin Mentor Graphicsin Inflexion UI -ohjelma, koska se oli Freescalen i.MX53-alustan tukemien ohjelmien joukossa. Ohjelma ei ollut minulle ennestään tuttu, joten opettelin sen käyttöä aluksi tekemällä Mentor Graphicsin Internet-sivulta löytyviä harjoituksia.

Työtä toteutettaessa pidettiin useita palavereita. Alkupalaverissa sovittiin eri vaiheiden tavoiteaikataulut, käyttöliittymän toiminnallisuus sekä alustavasti ulkoasu. Tällöin oli siis selvillä, miltä käyttöliittymän pitäisi näyttää ja miten sen pitäisi toimia, joten oli mahdollista aloittaa käyttöliittymän tekeminen. Myöhemmissä palavereissa sovittiin käyttöliittymään tehtävistä muutoksista, kuten graafisesta ulkoasusta, kuvakkeiden sijainnista ja toiminnallisuuden pienistä muutoksista. Suunnittelussa oli lähtökohtana asiakkaan toivomukset ja vaatimukset. Tärkeää oli myös ottaa huomioon laitteen kohderyhmä eli ikäihmiset.

7.1 Projektin, layoutien ja eventien luonti

Tässä luvussa on kursivoituna kohdat, joissa todellisuudessa käytettiin suunniteltuun käyttöliittymään kuuluvia nimiä.

Suunnittelun aluksi luotiin uusi projekti ja siihen perusnäky (Idle). Projekti luotiin seuraavasti:

- Valittiin Inflexion UI -ohjelmassa File -> New -> UI Design Project.
- Aukeavassa ikkunassa valittiin projektille *nimi* sekä projektin tyyppiksi base.
- Seuraavaksi valittiin next ja uudessa ikkunassa valittiin projektin Design size.

- Tämän jälkeen luotiin projekti Finish napilla.

Seuraavaksi Graphics-kansioon kopioitiin sillä hetkellä saatavilla olleet grafiikat. Sitten määritettiin käyttöliittymässä käytettävä taustakuva seuraavasti:

- UI Flow -kansio -> root.node -> Template&Background-lehti -> Background image -> Browse -> valittiin haluttu kuva -> OK.

Sen jälkeen luotiin perusnäkyä:

- Templates-kansio -> root.template -> Layout Manager -> Default layoutille.

Perusnäkyä luonnin jälkeen alettiin luoda muita layouteja käyttämällä Layout Managerin New Layout -toimintoa.

Uusien layoutien valmistuttua pystyttiin luomaan tarvittavat tapahtumat eli eventit. Eventit luotiin käyttämällä Layout Managerin Events-toimintoa. Alla on esimerkki yhden layoutin Eventin luonnista:

- Events-toiminnosta valitaan painalluksesta toimiva toiminto (New onTap Event).
- Events-toiminnon alle syntyy uusi painallustoiminto onTap.
- onTap-toiminnon Properties-valikkoon määritetään Element-kohdan Value-kenttään *arvo*.
- Layout Manageriin muuttuu nyt tekstiksi onTap *valittu arvo*, jonka jälkeen voidaan määrittää alapuolella olevan if-kohdan Properties-valikon Condition-kohdaksi TRUE.
- Nyt uusi toiminto siis on onTap *valittu arvo* ja sen alla if TRUE.
- If TRUE -kohdasta valitaan New Assignment, jolloin if TRUE -kohdan alapuolelle syntyy uusi toimeksianto (Assignment).
- Valitaan Set-kentän Properties-valikosta Field-kentän Value-arvoksi haluttu *layout* ja Propertyyn alta löytyvän Valuen arvoksi TRUE.
- Jotta luotu Event toimisi, pitää myös Element Managerissa tehdä muutoksia valittuun elementtiin.
- Valitaan Element Managerista *halutun elementin* Properties-valikko.
- Valitusta Properties-valikosta "In all layouts" -kohdan alta löytyvistä arvoista muutetaan Tappable-arvoksi full ja Tap tolerance -arvoksi 5.0.

- Lisäksi pitää määrittää root.templaten Settings-lehteen painallustoimintoa koskevien layoutien nimet ja näille datatyyppi (Data Type) -arvoksi boolean ja oletusarvoksi (Default Value) false, mikäli näitä arvoja ei ole jonkin muun painallustoiminnon yhteydessä määritetty.

Jotta luotu *event* toimisi halutulla tavalla, pitää uutta näkymää avattaessa sulkea näkyvillä ollut layout, joka tapahtuu määrittämällä *näkymän* arvoksi FALSE. Tämä tapahtuma määritetään myös luodun onTap *toiminnon nimi* if TRUE -painalluksen alle.

Layoutille luotiin samaan tyyliin myös muut tarvittavat eventit. Painallustoiminto muuttaa siis aina layoutien arvoksi TRUE/FALSE sen mukaan, mitä layouteja halutaan milloinkin olevan näkyvillä tai suljettuna.

7.2 Projektin liittäminen Eclipseen

Eclipsen käyttö oli välttämätöntä, jotta projekti voitiin liittää Freescalen i.MX53 Quick Start Board -alustalle. Projekti liitetään Eclipseen seuraavasti:

- Valitaan Eclipseessä File -> Import, jolloin aukeaa Import-ikkuna.
- Auenneessa ikkunassa valitaan General -> Existing Projects into Work space ja painetaan next.
- Tämän jälkeen Import-ikkunaan määritetään projekti, joka halutaan tuoda Eclipseen, ja painetaan Finish-nappia. Projekti on nyt liitetty Eclipseen.
- Eclipsen Package Explorer -näkymään tulee nyt näkyville liitetty projekti. Valitaan liitetty projekti ja siitä hiiren oikealla napilla Refresh.

Projektin pitäisi nyt olla liitettynä Eclipseen ilman virheitä. Jos virheitä ilmenee, ne voi yrittää poistaa seuraavasti:

- Valitaan projekti ja painetaan sitä hiiren oikealla napilla, auenneesta näkymästä valitaan Android Tools -> Fix Project Properties. Tämän pitäisi yleensä auttaa, jos liittämisen aikana on tullut joitain virheitä, sillä Eclipsen pitäisi pystyä ne näin korjaamaan.

7.3 Android-sovelluksen luonti ja liittäminen alustalle

Kun projekti oli saatu liitettyä Eclipseen, piti se vielä asentaa i.MX53-alustalle. Jotta asennus onnistuu, täytyy määrittää projektin ajoasetukset. Tämä tapahtuu seuraavasti:

- Valitaan Eclipsen Package Explorerissa oleva projekti ja siitä hiiren oikealla Run As -> Run Configurations. Tällöin aukeaa Run Configurations -ikkuna.
- Run Configurations -ikkunasta valitaan Android Application ja painetaan ylhäällä olevaa New launch configuration -nappia.
- Nyt voidaan määrittää uudet ajoasetukset. Android-lehdellä annetaan projektille nimi ja valitaan Browse-valinnalla projekti, jota halutaan ajaa. Muut valinnat tällä lehdellä jätetään ennalleen, eli jätetään täppä Launch Default Activity -kohtaan.
- Target-lehdelle määritetään Deployment Target Selection Mode -asetukseksi Manual, jonka jälkeen painetaan Apply.
- Varmistetaan vielä, että haluttu projekti on valittuna, ja painetaan Run. Tällöin aukeaa Android Device Chooser -ikkuna, jossa näkyy tietokoneeseen liitetty i.MX53-alusta, jonka nimenä on esimerkiksi 01234567890ABCD. Muita alustoja ei pitäisi näkyä, mikäli vain yksi ulkopuolinen alusta on liitetty tietokoneeseen.
- Valitaan se ja painetaan OK-nappia.
- Nyt haluttu ohjelma siirtyy i.MX53-alustalle ja käynnistyy siellä.
- Ohjelman voi tämän jälkeen aina käynnistää alustalta suoraan eikä alustan tarvitse tällöin enää olla tietokoneeseen kytkettynä. Mikäli ohjelmaan tehdään muutoksia, pitää se tietenkin päivittää alustalle uudestaan.

7.4 Testaus

Käyttöliittymän testaus on oleellinen osa käyttöliittymän suunnittelua. Tässä projektissa varsinaista testaussuunnitelmaa ei tehty, koska kyseessä oli vasta käyttöliittymän demon luonti. Toki projektin aikana tehtiin tietynlaista käyttöliittymän testausta. Ensinnäkin ongelmakohtia ja varsinaisia virheitä etsittiin aina kun uusia ominaisuuksia liitettiin tai entisiin ominaisuuksiin tehtiin muutoksia. Toiseksi demon toimintaa testattiin alustalla aina ennen sen toimittamista asiakkaalle.

Tällöin tarkastettiin pikaisesti, että toteutetut toiminnot toimivat, ja mikäli virheitä vielä ilmeni, ne korjattiin välittömästi. Myöhemmin, mikäli demon pohjalta lähdetään tekemään varsinaista tuotetta, on toki tehtävä testaussuunnitelma ja sen mukaiset testaukset. Varmaankin tällöin olisi myös käytettävyytestaus paikallaan.

8 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli selvittää käyttöliittymäsuunnittelun perusteita ja suunnitella Navicron Oy:n asiakkaan tilauksen mukainen graafinen käyttöliittymä Android-pohjaiseen laitteeseen. Käyttöliittymän piti toimia Freescalen i.MX53 -alustalla.

Navicron Oy:ssä päätettiin, että työ toteutettaisiin käyttämällä Mentor Graphicsin Inflexion UI -työkalua. Jotta työ saataisiin siirrettyä alustalle, käytettäisiin siinä Eclipseä. Näin toimittiin ja lopputuloksena saatiin kehitettyä tilattu käyttöliittymän demo, joka toimii asiakkaan vaatimusten mukaisesti. Kohderyhmä eli ikäihmiset otettiin erityisesti huomioon mahdollisimman helppojen valintanäkymien avulla.

Suunnittelutyöstä oli minulle hyötyä siten, että tutustuin minulle entuudestaan outoihin Inflexion UI- ja Eclipse-työkaluihin ja sain käyttää niitä käytännön työssä. Voin varmasti hyödyntää oppimaani tulevaisuudessa.

Jatkokehitystä ajateltaessa voidaan suunnitellun käyttöliittymän pohjalta tehdä toimiva laite, jossa myös toteutetaan kaikki toiminnallisuudet. Ajateltaessa Navicronin tulevaisuuden projekteja Freescalen i.MX53-alustan käytölle ei ole mitään estettä. Myös Inflexion UI:tä ja Eclipseä voidaan käyttää, mutta työkalujen valinta kannattaa tehdä aina tapauskohtaisesti.

LÄHTEET

1. Käyttöliittymä. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Käyttöliittymä>. Hakupäivä 4.11.2011.
2. Immonen, Jarkko 2003. Graafiset käyttöliittymät. Joensuu: Joensuun yliopisto, tietojenkäsittelytieteen laitos. Luentomoniste. Saatavissa: http://cs.joensuu.fi/~jimmonen/gkl_moniste/gkl_v202.html. Hakupäivä 4.11.2011.
3. Komentoliittymä. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Komentoliittymä>. Hakupäivä 4.11.2011.
4. FPC Command Line. 2011. Saatavissa: http://fi.wikipedia.org/wiki/Tiedosto:FPC_Command_Line.PNG. Hakupäivä 4.11.2011.
5. Tekstipohjainen käyttöliittymä. 2011. Saatavissa: http://fi.wikipedia.org/wiki/Tekstipohjainen_käyttöliittymä. Hakupäivä 5.11.2011.
6. Tekstipohjainen käyttöliittymä GNU Midnight Commander. 2011. Saatavissa: http://fi.wikipedia.org/wiki/Tiedosto:GNU_Midnight_Commander_4.1.36_Windows_Vista.png. Hakupäivä 5.11.2011.
7. Graafinen käyttöliittymä. 2011. Saatavissa: http://fi.wikipedia.org/wiki/Graafinen_käyttöliittymä. Hakupäivä 12.10.2011.
8. Puustinen, Johanna 2008. Windows 7 ensiesittelyssä. Saatavissa: http://www.tietoviikko.fi/kaikki_uutiset/windows+7+ensiesittelyssa/a151240. Hakupäivä 13.10.2011.

9. Lehto, Tero 2010. Intel ja Nokia tutkimaan 3d-käyttöliittymiä mobiililaitteissa. Saatavissa:
http://www.tietokone.fi/uutiset/intel_ja_nokia_tutkimaan_3d_kayttoliittymia.
Hakupäivä 5.11.2011.
10. Kinect. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Kinect>. Hakupäivä 5.11.2011.
11. Sinkkonen, Irmeli – Kuoppala, Hannu – Parkkinen, Jarmo – Vastamäki, Raino 2006. Käytettävyyden psykologia. 3., uudistettu painos. Helsinki: Edita Publishing Oy.
12. Sinkkonen, Irmeli 2004. Käyttöliittymät ja käytettävyys. Saatavissa:
<http://www.adage.fi/blogi/2004/kayttoliittymat-ja-kaytettavyys/>. Hakupäivä 24.8.2011.
13. Käytettävyys. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Kaytettavyys>. Hakupäivä 26.8.2011.
14. Nielsen, Jakob 1993. Usability Engineering. USA: Academic Press.
15. Kuitunen, Markku 2002. Tietojärjestelmän kehittäminen. Saatavissa:
<http://myy.helia.fi/~sys48d/Kaytettavyys/KtMat.htm>. Hakupäivä 2.9.2011.
16. Kone. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Kone>. Hakupäivä 29.8.2011.
17. Karvonen, Kristiina 2011. Käytettävyysuunnittelu. Saatavissa:
http://www2.uiah.fi/koulutuskeskus/media/kaytettavyysuunnittelu_karvone.pdf. Hakupäivä 16.10.2011.
18. Käytettävyydellä potkua tuotekehitykseen 2011. Työterveyslaitos Oulu. Opas pk-yrityksille. Saatavissa:
http://www.ttl.fi/ergonomia/menetelmat/Documents/kaytettavyydella_potku

a tuotekehitykseen.pdf. Hakupäivä 23.10.2011.

19. Käyttäjakeskeinen suunnittelu. 2011. Saatavissa:

http://fi.wikipedia.org/wiki/Käyttäjakeskeinen_suunnittelu. Hakupäivä 25.10.2011.

20. ISO 13407. 2011. Saatavissa: http://fi.wikipedia.org/wiki/ISO_13407. Hakupäivä 25.10.2011.

21. Android. 2011. Saatavissa: <http://fi.wikipedia.org/wiki/Android>. Hakupäivä 11.9.2011.

22. Android 2.2 perusnäkö. 2011. Saatavissa:

<http://fi.wikipedia.org/wiki/Tiedosto:Android-2.2.png>. Hakupäivä 30.9.2011.

23. Android 4.0 Google Nexus puhelimesta. 2011. Saatavissa:

http://fi.wikipedia.org/wiki/Tiedosto:Android_4.0.png. Hakupäivä 20.11.2011.

24. What is Android? 2011. Saatavissa:

<http://developer.android.com/guide/basics/what-is-android.html>. Hakupäivä 26.10.2011.

25. Android system architecture. 2011. Saatavissa:

<http://elinux.org/File:Android-system-architecture.jpg>. Hakupäivä 28.10.2011.

26. Mikä on Android?. 2011. Saatavissa: <http://blog.androidsuomi.fi/mika-on-android/>. Hakupäivä 28.10.2011.

27. Android Market. 2011. Saatavissa:

http://en.wikipedia.org/wiki/Android_Market/. Hakupäivä 29.10.2011.

28. Galaxy S Android Market. 2011. Saatavissa: http://www.mobiiliblogi.com/wp-content/uploads/2010/08/Galaxy_S_Android_Market.jpg. Hakupäivä

29.10.2011.

29. Lindholm, Christian 2000. Kuinka luoda asiantuntijoista tiimi? Teoksessa Keinonen, Turkka (toim.). Miten käytettävyys muotoillaan? Helsinki: Taideteollisen korkeakoulun julkaisu B 61.
30. Rouhiainen, Eeva-Kaisa 1997. Käyttöliittymän visuaalinen suunnittelu. Saatavissa: <http://www.mit.jyu.fi/opiskelu/seminaarit/bak/kayttoliittyma/>. Hakupäivä 27.10.2011.
31. Sinkkonen, Irmeli – Kuoppala, Hannu – Parkkinen, Jarmo – Vastamäki, Raiino 2009. Käytettävyyden psykologia. Saatavissa: http://www.adage.fi/uploads/pdf/Kaytettavyyden_psykologia.pdf. Hakupäivä 1.11.2011.
32. Käyttöliittymätestaus. 2011. Saatavissa: <http://www.oamk.fi/sbc/testaus/kayttoliittymatetaus.htm>. Hakupäivä 29.10.2011.
33. AUT. TT. Luento 3-19.11.2008. 2008. Saatavissa: <http://lipas.uwasa.fi/~timan/AUTO2020/auttt3.pdf>. Hakupäivä 4.11.2011.
34. Korhonen, Panu 2000. Käytettävyydesteistä liiketoiminnan ytimeen. Teoksessa Keinonen, Turkka (toim.). Miten käytettävyys muotoillaan? Helsinki: Taideteollisen korkeakoulun julkaisu B 61.
35. Inflexion UI for i.MX Processors. 2011. Saatavissa: <http://www.mentor.com/embedded-software/inflexion/ui-imx-processors>. Hakupäivä 16.8.2011.
36. Inflexion UI Solutions. 2011. Saatavissa: <http://www.mentor.com/embedded-software/events/esc/upload/inflexion-ui-esc-ds.pdf>. Hakupäivä 16.8.2011.

37. Inflexion UI Express. 2011. Saatavissa: <http://www.mentor.com/embedded-software/inflexion/>. Hakupäivä 16.8.2011.
38. Eclipse(IDE). 2011. Saatavissa: [http://fi.wikipedia.org/wiki/Eclipse_\(IDE\)](http://fi.wikipedia.org/wiki/Eclipse_(IDE)). Hakupäivä 18.8.2011.
39. Pitkänen, Jarmo 2010. Avoin ja monipuolinen sovelluskehitysympäristö. Saatavissa: http://www.tietokone.fi/softa/windows/eclipse_helios_3_6_1. Hakupäivä 18.8.2011.
40. IMX53QSB: i.MX53 Quick Start Board. 2011. Saatavissa: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=IMX53QSB. Hakupäivä 23.8.2011.
41. New low-cost Cortex A8 board from Freescale. 2011. Saatavissa: <http://www.linaro.org/linaro-blog/wp-content/uploads/2011/03/freescaleimx53quickstart.jpg>. Hakupäivä 23.8.2011.