



Mikko Määttä

GOLFOHJELMAN PROTOTYYPIN KEHITYS ANDROID-ÄLYPUHELIMELLE

GOLFOHJELMAN PROTOTYYPIN KEHITYS  
ANDROID ÄLYPUHELIMELLE

Mikko Määttä  
Opinnäytetyö  
Kevät 2012  
Tietotekniikan koulutusohjelma  
Oulun seudun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu  
Tietotekniikan koulutusohjelma, ohjelmistojen kehitys

---

Tekijä: Mikko Määttä

Opinnäytetyön nimi: Golfohjelman prototyypin kehitys Android-älypuhelimelle

Työn ohjaajat: Pertti Heikkilä (OAMK), Tuomo Lalli

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä: 49

---

Opinnäytetyössä toteutettiin sovellus golfin peluuta helpottavan ohjelman käyttöliittymän esittelykäyttöä varten. Työn tilaajana toimi Tuomo Lalli. Opinnäytetyössä tarkoituksena oli kehittää Tuomo Lallin ideasta ohjelma jota voisi esitellä sijoittajille. Ohjelman avulla luotaisiin mielikuvia sijoittajille ja asiakkaille tulevas-ta valmiista ohjelmistosta. Käyttäjä voisi täten myös kokeilla käyttöliittymän osia tulevan ohjelman toiminnallisuuksista suoraan älypuhelimessa eikä vain nähdä niistä kuvia.

Ohjelma suunniteltiin toimimaan kaikille Android-laitteille, joissa on käytössä vähintäänkin Android-alustan versio 2.3.3 (API 10) tai uudempi. Koska kysees-sä on esittelykäyttöön tuleva ohjelma, rajattiin ohjelmasta tarkoituksella van-hempien Android-alustan versioiden tuki pois. Ohjelma toteutettiin toimivaksi ZTE Blade -älypuhelimelle.

Opinnäytetyön toteutuksessa on käytetty Java-ohjelmointikieltä. Älypuhelimeen tulevan ohjelma käyttää Java-koodin Googlen kehittämiä Java-kirjastoja And-roid-alustalle. Työn tietoperustana sovellettiin pääasiassa Android Developer -sivuston sisältämiä oppeja. Opinnäytetyön teoriaosuudessa käytettiin lähdeai-neistoina pääasiassa verkkomateriaaleja, joista löytyy usein ajantasaista tietoa.

Opinnäytetyön tuloksena saatiin kehitettyä asiakkaan tilauksen mukainen proto-tyyppiohjelma. Prototyypiohjelma voi toimia pohjana myöhemmin mahdollisesti toteutettavalle ohjelmalle, mikäli ohjelma herättää sijoittajien mielenkiinnon.

---

Avainsanat: Android, golf, mobiilisovellukset

# ABSTRACT

Oulu University of Applied Sciences  
Information Technology, Software Development

---

Author: Mikko Määttä

Title of thesis: Development of Android Based Golf Programs Prototype

Supervisors: Pertti Heikkilä (OUAS), Tuomo Lalli

Term and year of completion: Spring 2012

Pages: 49

---

This thesis is about developing an application for demonstration purposes that makes playing golf easier. This thesis was commissioned by Tuomo Lalli. Idea behind this thesis was to develop an application that you could demonstrate to investors from idea by Tuomo Lalli. With the application you could create visions to investors and customers from the future complete application. User could also try the applications GUI at the smartphone himself and not just see pictures of it.

Application was planned to work with all Android devices, that use at least Android version 2.3.3 (API 10) or newer. Older version support was taken off because this is only an application to demonstrate what the future complete application could look like. Application was developed to work with ZTE Blade smartphone. The knowledge base of the work was mainly Android Developer webpage. Web-based materials were used as source of theoretical part of the thesis as digital sources contains the most up to date material.

As result of this thesis, an application was developed which fits the requirements of the commissioner. Application can be used as a base for future complete application if it awakes interest in investors.

---

Keywords: Android, golf, mobile software

## **ALKUSANAT**

Haluan kiittää Oulun seudun ammattikorkeakoulun tekniikan yksikköä ja työn tilaajaa Tuomo Lallia siitä, että teitte opinnäytetyöni tekemisestä mahdollista. Erityiset kiitokset kuitenkin Tuomo Lallille, joka ehdotti minulle tätä aihetta opinnäytetyöksi. Haluan lisäksi kiittää Pertti Heikkilää opinnäytetyön ohjauksesta sekä Tuula Hopeavuorta sen tekstin ohjauksesta, kiitokset myös Riitta Winterille opintojen ohjauksesta.

Oulussa 30.5.2012

Mikko Määttä

# SISÄLTÖ

TIIVISTELMÄ.....	3
ABSTRACT.....	4
ALKUSANAT .....	5
SISÄLTÖ.....	6
LYHENTEET JA TERMIT .....	8
1 JOHDANTO .....	9
2 VAATIMUSMÄÄRITTELY.....	10
2.1 Yleiskuvaus .....	10
2.2 Toiminnallisuuteen kohdistuvat vaatimukset .....	10
2.2.1 Toimintovaatimukset .....	10
2.2.2 Tekniset vaatimukset .....	12
3 ANDROID-KÄYTTÖJÄRJESTELMÄ .....	13
3.1 Arkkitehtuuri .....	14
3.1.1 Aktiviteetit.....	15
3.1.2 Käyttöliittymät.....	16
3.1.3 AndroidManifest-tiedosto .....	17
3.2 Google Play.....	19
3.3 Open Handset Alliance.....	20
3.4 Android-pohjaiset laitteet.....	21
4 TYÖSSÄ KÄYTETYT MENETELMÄT JA TYÖKALUT .....	23
4.1 Scrum-projektinhallintamenetelmä .....	23
4.1.1 Scrum-tiimi .....	23
4.1.2 Tapahtumat.....	24
4.1.3 Sprintti.....	25
4.1.4 Tuotokset .....	26
4.2 Versionhallinta.....	26
4.3 Työkalut.....	29
4.3.1 Eclipse.....	29
4.3.2 TortoiseSVN.....	31
4.3.3 GIMP .....	31
4.3.4 MediaWiki .....	32

5 TYÖN TOTEUTUS .....	33
5.1 Projektin aloitus .....	33
5.2 Projektin luonti .....	34
5.3 Projektin versionhallinta .....	34
5.3.1 Projektin päivittäminen versionhallintaan .....	35
5.3.2 Projektin päivittäminen uusimpaan versioon .....	36
5.4 Testaus .....	36
6 TULOKSET .....	41
7 YHTEENVETO .....	46
LÄHTEET .....	47

## LYHENTEET JA TERMIT

Android	Googlen kehittämä käyttöjärjestelmä mobiililaitteille.
API	Application Programming Interface eli ohjelmointirajapinta. Määritelmä, jonka mukaan eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja eli keskustella keskenään.
APK	Android Package eli Android-pohjaisen sovelluksen asennustiedoston muoto.
GUI	Graphical User Interface eli graafinen käyttöliittymä.
iOS	Applen kehittämä käyttöjärjestelmä mobiililaitteille.
iPad	Applen valmistama taulutietokone.
LCD	Liquid Crystal Display eli nestekidenäyttö. Nestekidenäytöt ovat laajalti käytössä matkapuhelimissa sekä elektroniikkalaitteissa.
PHP	Hypertext Preprocessor on laajalti käytetty ohjelmointikieli. Yleiskäyttöinen ohjelmistokieli, joka soveltuu eritoten Web-kehityksen käyttöön.
SMS	Short Message Service eli matkapuhelinten tekstiviestijärjestelmä. Käytetään lähettämään viestejä matkapuhelimesta toiseen.
Wiki	Verkkosivusto, jonka sisältöä käyttäjät voivat itse muokata joko kirjautumalla tai kirjautumatta sivustolle. Käytetään tiedonvälitykseen sekä yhteiskehittelyn alustana yrityksissä.
XML	Extensible Markup Language on merkintäkieli, joka määrittelee säännöt asiakirjoille, jotta se olisi sekä ihmisen että koneen luettava.



# 1 JOHDANTO

Golfin pelaajilla on tarve saada tietoa omasta pelisuorituksestaan golfkieroksella siten, että itse pelaajan ei tarvitsisi kiinnittää huomiota tiedon keruuseen. Kerättyä tietoa pelaaja voisi hyödyntää ohjaamaan harjoittelua sekä pelaaja voisi luoda omiin pelitaitoihinsa sopivan pelistrategian kullekin väylälle. Molemmilla asioilla on vaikutusta pelitulosten parantumiseen.

Tilaaajan tarkoituksena on tarjota älypuhelimien omistaville pelaajille kokonaisvaltainen golfharrastukseen liittyvä palvelu. Palvelussa pelaajan golfkierokselta kerättäisiin tietoa, kuten lyöntien määrä, lyöntipituudet mailoittain, lyöntialusta sekä osumatarkkuus, pelaajan kannalta mahdollisimman helposti.

Pelin aikana kerätystä tiedosta voitaisiin tarjota kooste pelituloksen parantamiseksi. Pelikieroksen aikana voitaisiin esimerkiksi tarjota pelaajalle hänen taitoihinsa ja pelattavaan kenttään sopivaa pelistrategia älypuhelimien sovelluksen avulla. Jokaiselle lyönnille voitaisiin ehdottaa myös sopivaa mailaa sekä tähtäyspistettä kenttäkartalta.

Tässä opinnäytetyössä keskityttiin prototyypiohjelman käyttöliittymän toteuttamiseen esittelykäyttöön, jotta idean voisi esittää konkreettisesti sijoittajille. Työn tilaajana toimi Tuomo Lalli, jolla on aikeena perustaa oma yritys. Hänen tarkoituksenaan on kehittää valmis ohjelmisto omassa liiketoiminnassaan, jos prototyyppi osoittaa sijoittajille ohjelmiston tarpeen ja kehitysmahdollisuudet. Projektin aikana yritystä kehitettiin Yritystakomon kautta, joka on innovatiivinen toimitila ja -paikka kaikille uusien yritys- ja tuoteideoiden kehittämisestä kiinnostuneille (1). Sovellusta varten valittiin Android-alustalla toimiva älypuhelin, koska siihen löytyi valmis testilaitteisto ja ohjelmointikemusta.

## **2 VAATIMUSMÄÄRITTELY**

Vaatimusmäärittely on ohjelmistoprojektin kulmakivi, jossa kuvataan ohjelmiston tavoitteet ja vaatimukset. Siinä määritellään tuleva lopullinen ohjelmisto sekä sen toiminnallisuudet että keinot, kuinka nämä toiminnallisuudet tulisi saavuttaa. Vaatimusmäärittelyssä voidaan myös määrittää laatuvaatimuksia, kuten esimerkiksi ohjelmiston käytettävyys, luotettavuus ja turvallisuus sekä ylläpidettävyys.

### **2.1 Yleiskuvaus**

Opinnäytetyön lopputuotteena syntyvän sovelluksen tulee toimia tätä projektia varten valitulla Android-alustalla toimivalla ZTE Blade -älypuhelimella. Sovelluksessa tulee olla näkymä tilanteesta, jossa pelaaja on väylällä ja aloittaa pelaamisen, sekä näkymä viheriöltä. Sovelluksen tulee antaa esimerkkejä toiminnallisuuksista, joita ohjelmalla voidaan toteuttaa, ja sen avulla voitaisiin synnyttää mielikuvia sijoittajille. Sovelluksen tulee mahdollistaa yhden golfväylän virtuaalinen pelaaminen.

### **2.2 Toiminnallisuuteen kohdistuvat vaatimukset**

Tässä luvussa määritellään ohjelman toiminnallisuudet. Toiminnallisuuteen kohdistuvilla vaatimuksilla määritetään, mitä lopullisen ohjelmiston tulee sisältää, sekä miten ohjelmiston tulee toimia. Vaatimukset on laatinut työn tilaaja Tuomo Lalli.

#### **2.2.1 Toimintovaatimukset**

Golfin pelaaja on pelikentällä valmistautumassa lyöntiin ja haluaa apua pelistrategiassaan. Pelaaja valitsee älypuhelimessaan olevasta ohjelmasta näkymän, joka näyttää pelattavan väylän kenttäkartan. Kartalla näkyy pelaajan tämänhetkinen sijainti sekä karttamateriaalin perusteella oletettu lyöntialusta

(tee, fairway, rough, semi rough, foregreen, green). Pelaaja voi myös halutessaan vaihtaa lyöntialustan tyyppiä sekä mailavalintaa.

Kartalla näkyy tähtäyspiste, johon pelaajan lyönnin tulisi suuntautua, sekä etäisyys pelaajasta tähtäyspisteeseen. Kyseessä on pelaajan ensimmäinen pelitapahtuma, joten tähtäyspiste määräytyy oletusarvoisesti väylän keskelle. Matkan ollessa alle 200 m viheriölle asetetaan tähtäyspiste sen keskelle. Pelaajan tulee voida liikuttaa tähtäyspistettä haluamaansa kohtaan.

Ohjelma näyttää kenttäkartalla ehdotetut seuraavat lyöntipaikat pohjautuen tähtäyspisteeseen. Tähtäyspistettä tarpeeksi siirrettäessä ohjelma näyttää uudet ehdotetut lyöntipaikat. Jokaiselta lyöntipaikalta näytetään ehdotus käytettävästä mailasta. Tämän lisäksi tähtäyspisteen viereen kuvataan alue, johon pelaajan lyönti todennäköisesti sijoittuu. Ohjelmassa tulee myös olla pieni todennäköisyys, että lyönti sijoittuu alueen ulkopuolelle. Mikäli tähtäyspiste viedään yli 235 m:n päähän pelaajasta, ei mailavalinnan tai lyöntipaikojen ehdotelmia näytetä.

Ohjelmassa lyönnin tulee tapahtua heilauttamalla älypuhelinta. Ohjelma päivittää lyöntitapahtuman jälkeen ehdotetut lyöntipaikat sekä mailavalinnan pelaajan uuden sijainnin perusteella. Pelaajan saapuessa viheriölle näytetään enää pelkästään viheriön näkymä.

Viheriön näkymässä tähtäyspistettä ei enää näytetä, vaan tähtäyspisteenä käytetään viheriöllä näkyvää lippua. Lyönnin tulee tapahtua myös viheriöllä älypuhelinta heilauttamalla. Ohjelma arpoo pelaajan lyönnin jälkeisen sijainnin sekä tarkistaa, onko pelaaja saanut pallon viheriön reikään. Ohjelman tulee kirjata ylös lyöntipaikat sekä niillä käytetyt mailat ja lyöntialustan tyypit. Nämä tiedot ilmoitetaan ohjelman siirtyessä loppunäkymään, kun pelaaja on saanut pallon viheriön reikään.

## 2.2.2 Tekniset vaatimukset

Projektin alussa ei ollut tarkkoja teknisiä vaatimuksia, vaan niitä luotiin pitkin projektin etenemistä, koska tilaajalla ei ollut kokemusta ohjelmistosuunnittelusta Android-alustalle. Tämän vuoksi teknisen tason vaatimukset muuttuivat projektin edetessä useasti. Käyttöliittymäsuunnitelmaa toteutettaessa huomattiin useassa kohden, että jotakin asiaa ei kannattanutkaan toteuttaa niin kuin se oli suunniteltu. Tällaisissa tilanteissa päätettiin muuttaa ongelmatoimintoa toteutuksen kannalta sopivammaksi.

Vaatimuksena oli, että kaikkien toimintojen ja ulkoasun tulisi olla mahdollisimman selkeitä toiminnallisuuksiltaan käyttäjälle. Tämä vaatimus toimi niin ikään sääntönä ja apuna ohjelman ulkoasua muokattaessa sekä toimintoja toteutettaessa. Teknisen tason vaatimuksia oli projektin luonteen vuoksi mahdollista muokata. Halutut muutokset hyväksyttiin tilaajalla.

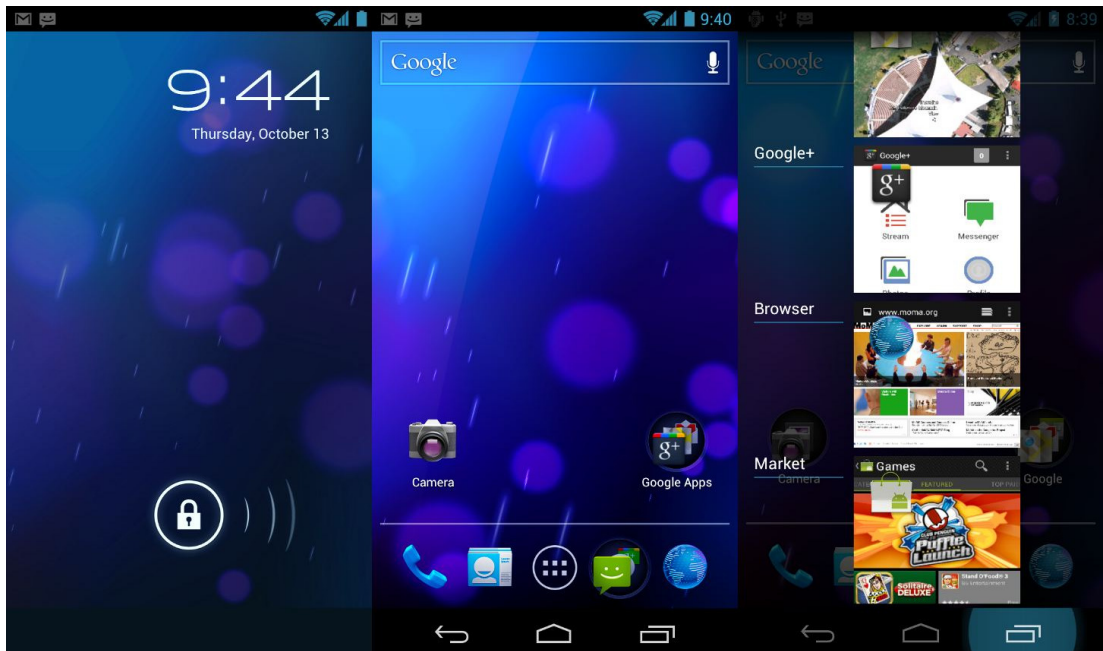
### 3 ANDROID-KÄYTTÖJÄRJESTELMÄ

Android on Linux-pohjainen käyttöjärjestelmä mobiililaitteille, kuten älypuhelimille ja taulutietokoneille. Androidin kehitti alun perin Android Inc, joka on perustettu lokakuussa 2003 ja jota johti Andy Rubin. Yritys kehitti käyttöjärjestelmänsä salassa paljastaen ainoastaan kehittävänsä ohjelmistoa matkapuhelimille. Yritys kuitenkin joutui rahavaikeuksiin, jonka jälkeen Google osti Android Inc:n syyskuussa 2005. Android Inc:n päähenkilöt kuten Andy Rubin, Rich Miner ja Chris White jäivät yhtiön palvelukseen yrityskaupan jälkeen. (2.)

Androidia kehittää tällä hetkellä Open Handset Alliance. Open Handset Alliance perustettiin 6. marraskuuta 2007 ja samana päivänä julkistettiin Android. Kehitystyökalut julkaistiin nopealla aikataululla jo 12. marraskuuta 2007. (2.)

Ensimmäinen virallinen myynnissä oleva puhelin, joka käytti Android-pohjaista käyttöjärjestelmää, oli HTC Dream (T-Mobile G1). Puhelin julkaistiin 22. lokakuussa 2008 ja se käytti Android-versiota 1.0. Viimeisen julkaistu Android-versio, joka tukee sekä puhelimia että taulutietokoneita, on 4.0 (Ice Cream Sandwich, ICS) (kuva 1). (2.)

Androidin suosio on kasvanut viime vuosina hurjasti. Android nousi Nokian Symbianin ohi älypuhelimien myydyimmäksi mobiilialustaksi 2010 viimeisellä neljänneksellä vain kolme vuotta julkistamisen jälkeen (3). Uutistoimisto Gartner arvioi Android-pohjaisten älypuhelimien markkinaosuudeksi vuoden 2011 kolmannella neljänneksellä huikeat 52,5 % (4).



KUVA 1. Android 4.0, Ice Cream Sandwich (5)

### 3.1 Arkkitehtuuri

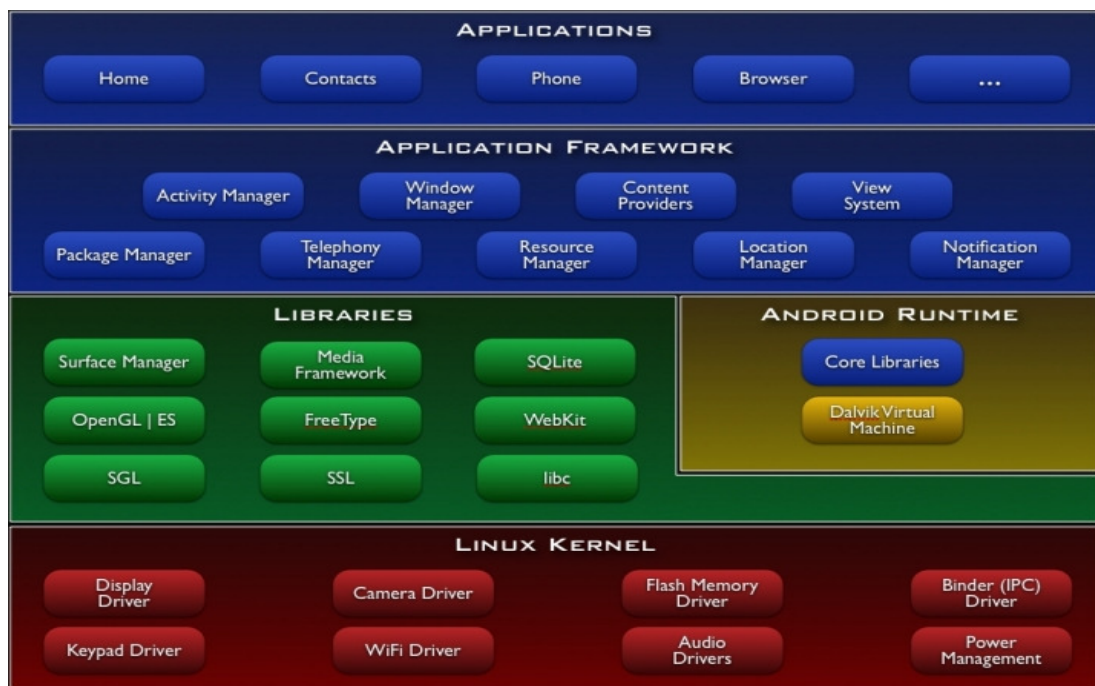
Android on mobiililaitteille suunniteltu ohjelmistopino, joka sisältää käyttöjärjestelmän. Androidin arkkitehtuuri (kuva 2) muodostuu viidestä eri tasosta. Ylimpänä arkkitehtuurissa ovat valmiit sovellukset, kuten sähköposti- ja SMS-ohjelma, kalenteri, kartat, selain ja yhteystiedot. Tälle tasolle asetetaan myös ohjelmistokehittäjän luomat sovellukset. (6.)

Toisena on sovelluskehitystaso (Application Framework), joka tarjoaa ohjelmistokehittäjille valmiita komponentteja. Ohjelmistokehittäjät voivat käyttää hyödyksi laitteistoa, päästä käsiksi paikannustietoihin, suorittaa taustaohjelmia, asettaa hälytyksiä, lisätä ilmoituksia ilmoituspalkkiin ja paljon muuta. (6.)

Kolmantena ovat kirjastot (Libraries) jotka ovat C- ja C++-kirjastoja. Kirjastoista löytyvät esimerkiksi mediakirjastot, mistä löytyy tuki monille suosituille toisto, sekä nauhoitusmahdollisuus useille ääni- ja videoformaateille, kuten esimerkiksi MPEG4, H.264 ja MP3. Näihin kirjastoihin on mahdollista päästä käsiksi suoraan sovelluskehitystasolta. (6.)

Neljäntenä on Android Runtime, jossa tapahtuu sovellusten varsinainen suorittaminen. Täällä myös sijaitsevat tärkeimmät kirjastot kuten Core Libraries. Androidin jokainen sovellus suoritetaan omana prosessina Dalvik Virtual Machinen avulla. Dalvik on kirjoitettu siten, että se voi ajaa useaa virtuaalikonetta tehokkaasti yhtä aikaa. (6.)

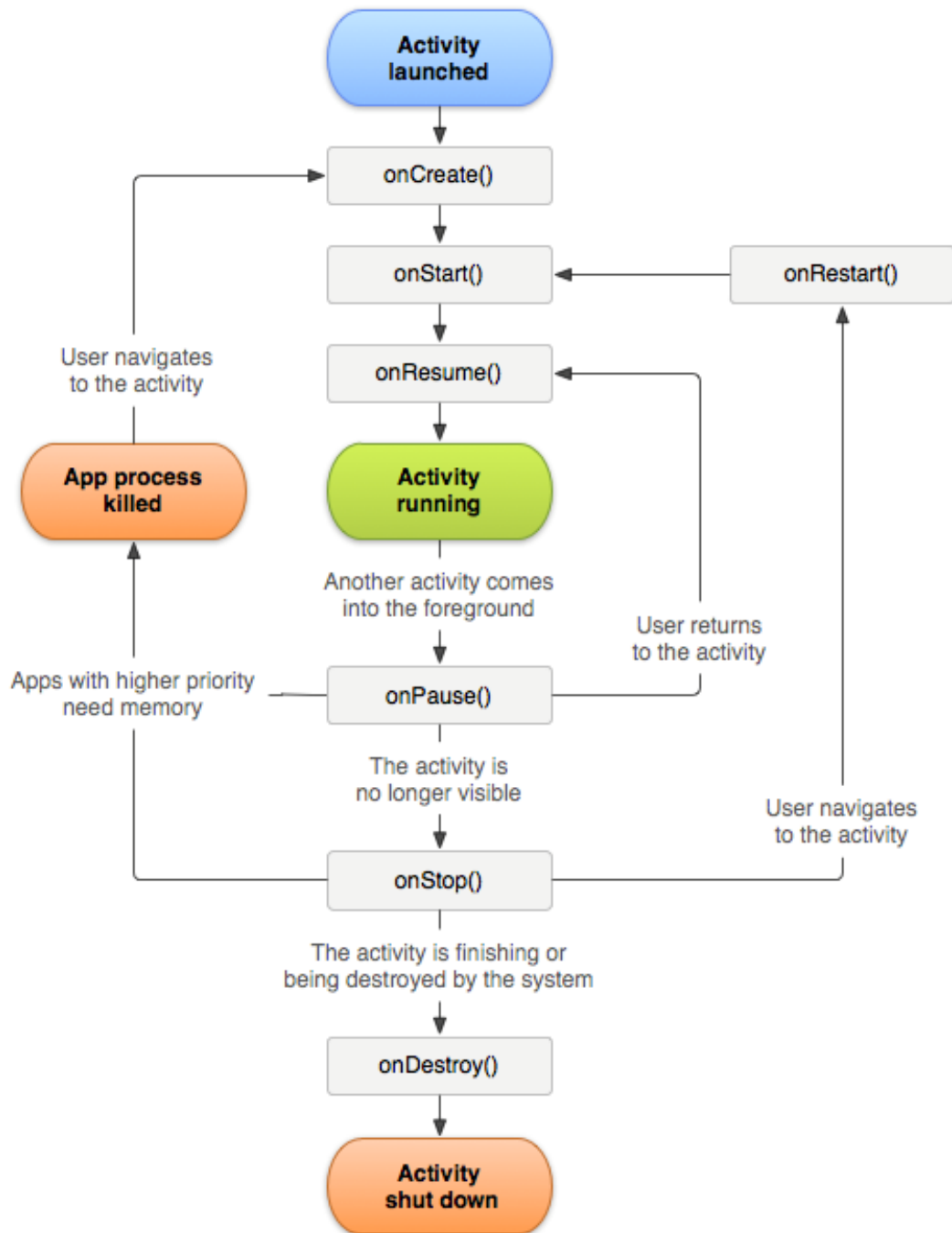
Viimeisenä, mutta tärkeimpänä tasona on Linuxin käyttöjärjestelmän ydin (Linux Kernel), jota käytetään vain tärkeimpiin toimintoihin kuten turvallisuuden, muistin ja prosessin hallintaan sekä verkkopinnoon ja ajureihin. Käyttöjärjestelmän ydin toimii myös abstraktiorajapintana laitteiston ja loppujen ohjelmistojen välillä. (6.)



KUVA 2. Android-arkkitehtuuri (7)

### 3.1.1 Aktiviteetit

Aktiviteetti on näkymä käyttäjälle. Aktiviteettien kautta käyttäjä on vuorovaikutuksessa laitteiston kanssa. Ohjelmisto yleensä koostuu usein useista aktiviteeteista. Jokaisella aktiviteetilla on elinkaarensa (kuva 3). Kaikki aktiviteetit tulee määritellä AndroidManifest.xml-tiedostossa. (8.)



KUVA 3. Androidin aktiviteetin elinkaari (9)

### 3.1.2 Käyttöliittymät

Android-ohjelman tyylisääntöjen mukaan käyttöliittymä- ja ohjelmakoodi pyritään pitämään erillään toisistaan. Tarpeen vaatiessa voidaan kuitenkin luoda



käyttöliittymäkomponentteja suoraan ohjelmistokoodissa. Käyttöliittymäkoodi sijaitsee layout-kansiossa XML-tiedostoissa (kuva 4).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <ImageButton
        android:id="@+id/btnTerrain"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:background="@null"
        android:onClick="showTerrainMenu"
        android:scaleType="fitXY"
        android:src="@drawable/terrain_tee"
        android:contentDescription="@string/btnTerrain" />

    <ImageButton
        android:id="@+id/btnClubs"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:background="@null"
        android:onClick="showClubsMenu"
        android:scaleType="fitXY"
        android:src="@drawable/club_driver"
        android:contentDescription="@string/btnClubs" />

</RelativeLayout>
```

*KUVA 4. Esimerkki ohjelman layout-tiedostosta*

### 3.1.3 AndroidManifest-tiedosto

AndroidManifest.xml-tiedosto (kuva 5) sisältää välttämättömät tiedot ohjelmistosta Android-järjestelmälle. Se sisältää muun muassa tiedot ohjelman käyttämistä aktiviteeteista sekä oikeuksista (kuva 6), kuten internet, SD-kortin käyttö, paikannustiedot ja kohdealueista. Kohdealueen version minimivaatimus ilmoitetaan manifest-tiedostossa omalla rivillä (kuva 7). Ohjelman versionumerointi tapahtuu myös tässä tiedostossa. (10.)

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.caddieon.demo"
    android:versionCode="1"
    android:versionName="0.2" >

    <uses-sdk android:minSdkVersion="10" />

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" android:theme="@android:style/Theme.Light">
        <activity
            android:name=".DemoActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

*KUVA 5. Esimerkki AndroidManifest-tiedostosta*

```

<uses-permission android:name="android.permission.INTERNET"/>

```

*KUVA 6. Esimerkki ohjelman oikeuksien määrittämisestä*

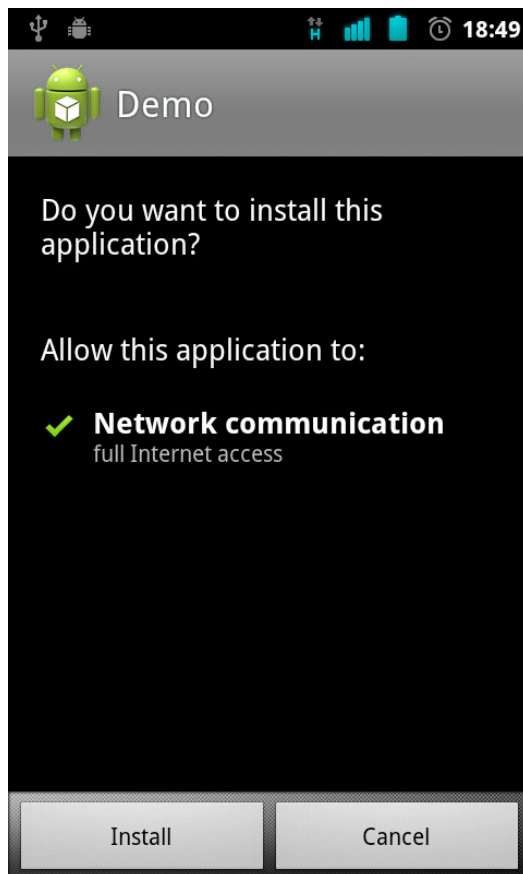
```

<uses-sdk android:minSdkVersion="10" />

```

*KUVA 7. Esimerkki kohdealustan minimiversion vaatimuksen määrittämisestä*

Kuvan 6 esimerkissä kerrotaan, että ohjelma tarvitsee oikeuden Internet-yhteyden käyttöön. Kun ohjelmaa asennetaan, ilmoittaa ohjelma kaikki sen tarvitsemat oikeudet (kuva 8). Tämä näkymä tulee aina ohjelmistoa asennettaessa ja siinä lukee kaikki ohjelman vaatimat oikeudet. Käyttäjä voi peruuttaa asennuksen painamalla Cancel-painiketta jos hän ei halua hyväksyä ohjelman vaatimia oikeuksia.



*KUVA 8. Ohjelmaa asentaessa ilmoitetaan mitä oikeuksia kyseinen ohjelma tarvitsee toimiakseen*

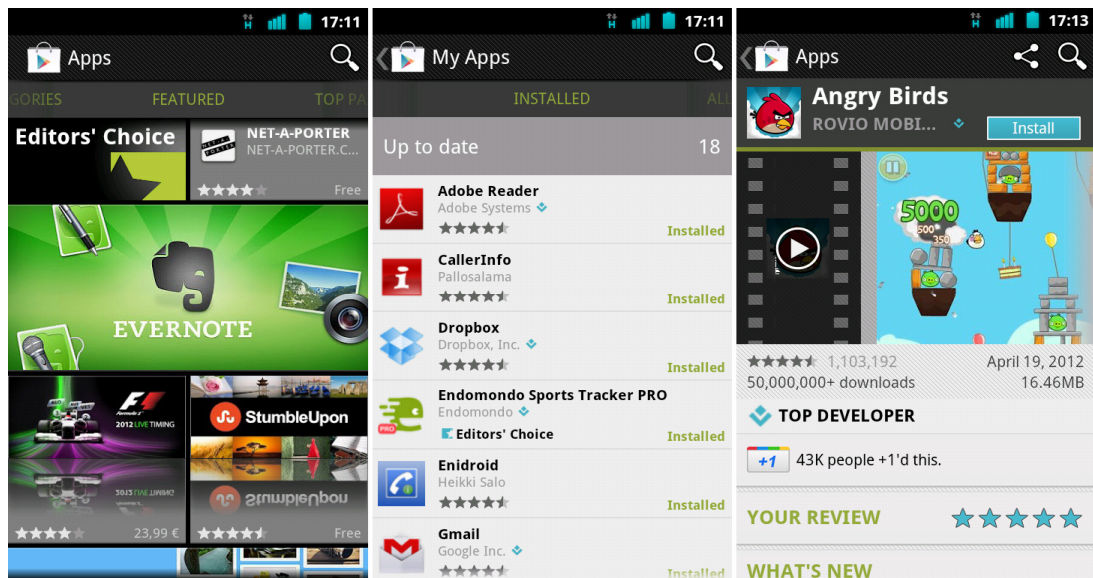
### 3.2 Google Play

Google Play (kuva 9) on verkkokauppa musiikille, elokuville, kirjoille ja Android-ohjelmille ja -peleille. Nimensä mukaan Google Play on Googlen kehittämä. Google Play vaatii Google-tilin. Google Playn avulla voi ladata ja asentaa uusia ohjelmia. Myös vanhojen ohjelmien päivitys Android-pohjaisissa laitteissa suoritetaan tämän kautta. Laittevalmistajat ovat yleensä esiasentaneet Google Play- tai vanhan Android Market -sovelluksen valmistamiinsa tuotteisiin. Vanha Android Market päivittyy automaattisesti Google Playksi.

Google Playhin liittyminen maksaa ohjelmistokehittäjälle 25 USD eli noin 19 euroa, mutta rekisteröityminen on kertaluontoinen ja elinikäinen. Ohjelmistokehittäjät saavat 70 % ohjelmiston hinnasta ja loput 30 % jaetaan jakelijan ja maksun käsittelijän kanssa. Google itse ei tienaa prosenttiakaan, vaan tuotto

Android Marketista maksetaan kehittäjille Google Checkout Merchant -tileille tai joissakin maissa Google AdSensen kautta. (11.)

Google Playssä voidaan julkaista maksuttomia ja maksullisia ohjelmistoja. Yleensä ilmaisessa sovelluksessa näytetään mainoksia tai se on ominaisuuksiltaan rajoittuneempi kuin maksullinen. Google Playstä löytyi yli 450 000 ohjelmistoa helmikuussa 2012 ja 10 miljardin latauksen raja rikottiin joulukuussa 2011 (12).



KUVA 9. Google Play -mobiilisovelluksen näkymä ZTE Blade -puhelimessa

### 3.3 Open Handset Alliance

Open Handset Alliance -liitto perustettiin 5. marraskuuta 2007. Liittoon kuului alun perin Google ja 34 yritystä joihin kuului mobiililaitteen valmistajia, ohjelmiston kehittäjiä ja joitain tukiasemien valmistajia. (13.)

Open Handset Alliance on nykyään 84 teknologia- ja mobiiliyrityksen liitto. Liitto on perustettu nopeuttamaan innovaatioita mobiililaitteissa ja tarjoamaan kuluttajille toiminnallisuudeltaan rikkaampia, halvempi ja parempia mobiililaitteen käyttökokemuksia (14). Liiton tarkoituksena on kilpailla muita mobiilialustoja vastaan kuten Apple IOS, Microsoft Windows Phone, Research In Motion (RIM) Blackberry ja Samsung Bada.

### 3.4 Android-pohjaiset laitteet

Android on tällä hetkellä maailman myydyin mobiilialusta. Android-pohjaisia älypuhelimia kehittävät useat suuret valmistajat. Näihin valmistajiin lukeutuvat esimerkiksi Samsung, HTC, LG ja Sony-Ericson sekä joukko pienempiä valmistajia kuten ZTE. (15.)

Jotkut puhelimen valmistajat käyttävät niin sanottua vakio-Androidia (Vanilla), mutta suurin osa valmistajista on lisännyt puhelimeensa oman käyttöliittymänsä ja tarjoaa myös omat oletussovelluksensa. Android-pohjaisten älypuhelimien hintaluokka vaihtelee halvoista, noin 100 euron puhelimista, kalliimpiin 700 euron älypuhelimiin asti. Merkittävin ero hinnan lisäksi on suorittimen ja grafiikkaprosessorin laskentanopeudessa sekä näytön koossa ja tarkkuudessa. Yleensä myös kalliimpien mallien kuorien materiaalit ovat kestävämpiä sekä kameran optiikat ovat parempia. Silti kaikki Android-pohjaiset puhelimet ovat pohjimmiltaan älypuhelimia ja niitä voidaan käyttää peruskäytön lisäksi netin selaamiseen, kuvien ja videoiden katseluun sekä musiikin soittamiseen. (16.)

Taulutietokoneet eli tabletit ovat lyömässä itseään läpi. Vähän aikaa sitten ei tableteille löytynyt käyttötarkoituksia ja niiden suosio oli huonoa. Tabletteja on valmistettu useiden vuosien ajan ja ne ovat tulleet suosituksi vasta Applen valmistaman ensimmäisen iPadin myötä. Tableteissa Android ei ole vielä saanut lyötyä itseään läpi, vaan Apple johtaa kilpailua iOS-pohjaisella iPadillaan (17).

Tabletteja käytetään nykyään paljon netissä surffailuun, elokuvien katsomiseen, sähköpostien lukemiseen sekä jopa työntekoon. Erityisesti graafiset suunnittelijat ovat ihastuneet näihin suuren kosketusnäytön sisältäviin puoliksi tietokoneisiin ja puoliksi kännyköihin. Jopa lentäjät käyttävät nykyään tabletteja. (18.)

Android-pohjaiset laitteet ovat vahvasti sidoksissa Googlen palveluihin, ja laitteen ominaisuuksien täydelliseen hyödyntämiseen tarvitaan käyttäjätili

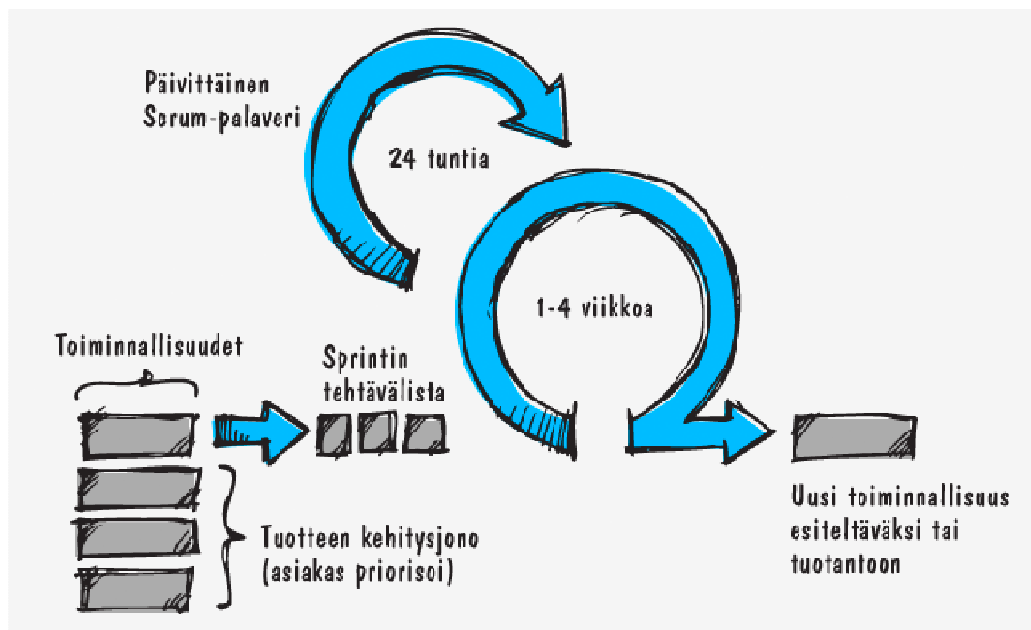
Googlen palveluihin. Android-pohjaista laitetta on myös mahdollista käyttää ilman Googlen palveluita, mutta tämä rajoittaa laitteen ominaisuuksia rajusti.

## 4 TYÖSSÄ KÄYTETYT MENETELMÄT JA TYÖKALUT

Tässä luvussa kerrotaan opinnäytetyön aikana käytetyistä projektin- ja versiohallintamenetelmistä sekä työkaluista. Kaikki projektissa käytetyt menetelmät ja työkalut valittiin yhdessä tilaajan kanssa.

### 4.1 Scrum-projektinhallintamenetelmä

Scrum on ketterä prosessikehys (kuva 10), jonka käytännöt luovat erinomaisen näkymän ohjelmistokehityksen todelliseen edistymiseen. Scrumia käytettäessä pystytään seuraamaan ohjelmistokehityksen etenemistä. Ajankohittaisen tiedon avulla voidaan tehdä koko kehitystä koskevia päätöksiä, parantaa toimintatapoja ja selvittää kehitystä hidastavat asiat (19).



KUVA 10. Scrum-prosessin elinkaari (20)

#### 4.1.1 Scrum-tiimi

Scrum-tiimi koostuu tuotteen omistajasta, kehitystiimistä ja Scrum-vetäjästä. Scrum-tiimit ovat itseohjautuvia eli tiimit päättävät parhaaksi katsomallaan tavalla miten työ tehdään ilman ulkoista ohjausta. (21, s. 4.)

Tuotteen omistaja on vastuussa tuotteen kehitystiimin tekemän työn arvon maksimoinnista sekä tuotteen kehitysjonosta. Tuotteen omistajan velvollisuutena on kehitysjonon kohtien selkeä kirjallinen ilmaisu, kohtien järjestäminen siten, että tavoitteet saavutetaan parhaalla mahdollisella tavalla, kehitystiimin työn arvon ja maksimoinnin varmistaminen sekä tuotteen kehitysjonon avoimuuden, läpinäkyvyyden ja ymmärrettävyyden varmistaminen. Tuotteen omistajan tulee myös käydä kehitysjono läpi kehitystiimin kanssa ja varmistaa, että kehitystiimi ymmärtää kohdat riittävällä tarkkuudella. (21, s. 4–5.)

Kehitystiimi koostuu työntekijöistä, joiden tehtävänä on kehittää kehitysjonon sisällöstä julkaisukelpoinen tuoteversio jokaisessa sprintissä. Ainoastaan kehitystiimin jäsenet osallistuvat tuoteversion kehitykseen. Kehitystiimin vastuu kuuluu koko kehitystiimille yhdessä. Kehitystiimit eivät sisällä alatiimejä, joiden tehtävänä olisi vastata tietyistä osa-alueista kuten testaaminen. (21, s. 5–6.)

Scrum-vetäjän tehtävä on vastata siitä, että kaikki ymmärtävät ja osaavat käyttää Scrumia. Scrum-vetäjän tulee osata palvella eri osia Scrum-tiimistä. Hänen pitää osata toimia tuotteenomistajan kannalta selkeästi sekä opastaa tuotteenomistajaa kehitysjonon tehokkaaseen hallintaan sekä kertoa kehitystiimille selkeästi tuotteen visiot, tavoitteiden sekä kehitysjonon kohdat. (21, s. 6-7.)

#### **4.1.2 Tapahtumat**

Scrumissa on käytössä ennalta sovittuja tapahtumia, jotka on luotu tuomaan säännöllisyyttä ja minimoimaan muiden kuin Scrum-palavereiden tarve. Scrumin tapahtumille määritetään tietyt aikarajat kuten maksimipituus. Tämän avulla varmistetaan, että suunnittelulle varataan tarpeeksi aikaa eikä suunnitteluprosessissa pääse syntymään hukka-aikaa. (21, s. 7.)

Kaikissa Scrumin tapahtumissa voidaan tarkastella ja sopeuttaa jotain. Ainoa poikkeus on itse sprintti, joka sisältää tapahtumat. Tapahtumien ennalta



suunnittelu lisää tuotekehitykselle äärimmäisen tärkeää läpinäkyvyyttä ja mahdollistaa tapahtuman tarkastelun. Yhdenkin tapahtuman pois jättäminen vähentää Scrum-projektin läpinäkyvyyttä ja johtaa menetettyyn tilaisuuteen tarkastelussa sekä sopeuttamisessa. (21, s. 7.)

### **4.1.3 Sprintti**

Scrumin avulla voidaan rakentaa ohjelmistokehitys erimittaisten syklien ympärille. Scrumin ytimenä toimii sprintti. Sprintillä tarkoitetaan yhtä kehitysjaksoa, jonka jälkeen tuotteen pitäisi olla julkaisuvalmis. Sprintin kesto voi vaihdella viikosta aina kahteen kuukauteen, mutta sprintin pituus pysyy samana koko kehityksen ajan. (21, s. 7; 21.)

Sprinttien ideana on aina tietyn tavoitteen saavuttaminen. Jokainen sprintti sisältää määritelmän siitä mitä ollaan toteuttamassa, suunnitelman, mikä ohjaa toteutusta, sekä työn tuloksena syntyvän tuoteversion. Sprintti käsittää sprintin aikana käytävät asiat kuten suunnittelupalaverin, päiväpalaverin, kehitystyön ja sprinttikatselmuksen. (21, s. 7.)

Sprintit aloitetaan suunnittelupalaverilla. Suunnittelupalaverissa käydään läpi koko Scrum-tiimin kesken asiat mitä tullaan toimittamaan alkavan sprintin tuoteversiossa ja miten siihen tarvittava työ voitaisiin toteuttaa. (21, s. 8–9.)

Sprintin aikana on hyvä pitää päiväpalavereita. Päiväpalaveri on tapahtuma, jossa käydään läpi mitä on saatu aikaan viime tapaamisen jälkeen ja mitä ollaan tekemässä ennen seuraavaa tapaamista. Päiväpalavereissa voidaan myös käydä läpi esteitä joita on ilmennyt palavereiden välissä. (21, s. 9–10.)

Sprintin lopussa pidetään aina sprinttikatselmus. Katselmoinnissa selvitetään, mikä osa työssä on valmista ja mikä on vielä puutteellinen sekä käydään läpi sprintin aikana esille tulleet ongelmat ja se kuinka niistä on selvitty. Tuotteen omistaja arvioi kehitysjonon ja aikataulun tilanteen sekä pohditaan koko ryhmän kanssa mitä seuraavaksi pitäisi ja kannattaisi tehdä, jotta katselmus antaisi hyvän pohjan seuraavalle suunnittelupalaverille. Katselmuk-

sen tuloksena syntyy tarkistettu tuotteen kehitysjoono, joka sisältää alustavat tuotteen kehitysjonot seuraavalle sprintille. (21, s. 10.)

Sprintin aikana ovat käytössä tarkat ennalta määritetyt säännöt. Sprintin aikana ei voida enää tehdä muutoksia, jotka vaikuttaisivat sprintin tavoitteeseen. Laatutavoitteita ei lasketa, eikä kehitystiimin koostumusta voida muuttaa. Sprintin aikana kuitenkin sisältöä voidaan tarkentaa ja sen muutoksista voidaan neuvotella tuoteomistajan sekä kehitystiimin välillä, kun tiedetään lisää ratkaistavasta ongelmasta. (21, s. 7–8.)

#### **4.1.4 Tuotokset**

Valmiin tuotteen määritelmä käydään läpi Scrum-tiimin kanssa, jotta jokaisella on ymmärrys siitä, mitä valmiilla tarkoitetaan. Valmiin määritelmä voi vaihdella huomattavasti eri Scrum-tiimien välillä ja siksi onkin tärkeää, että kaikilla on sama näkemys asiasta. (21, s. 11–12.)

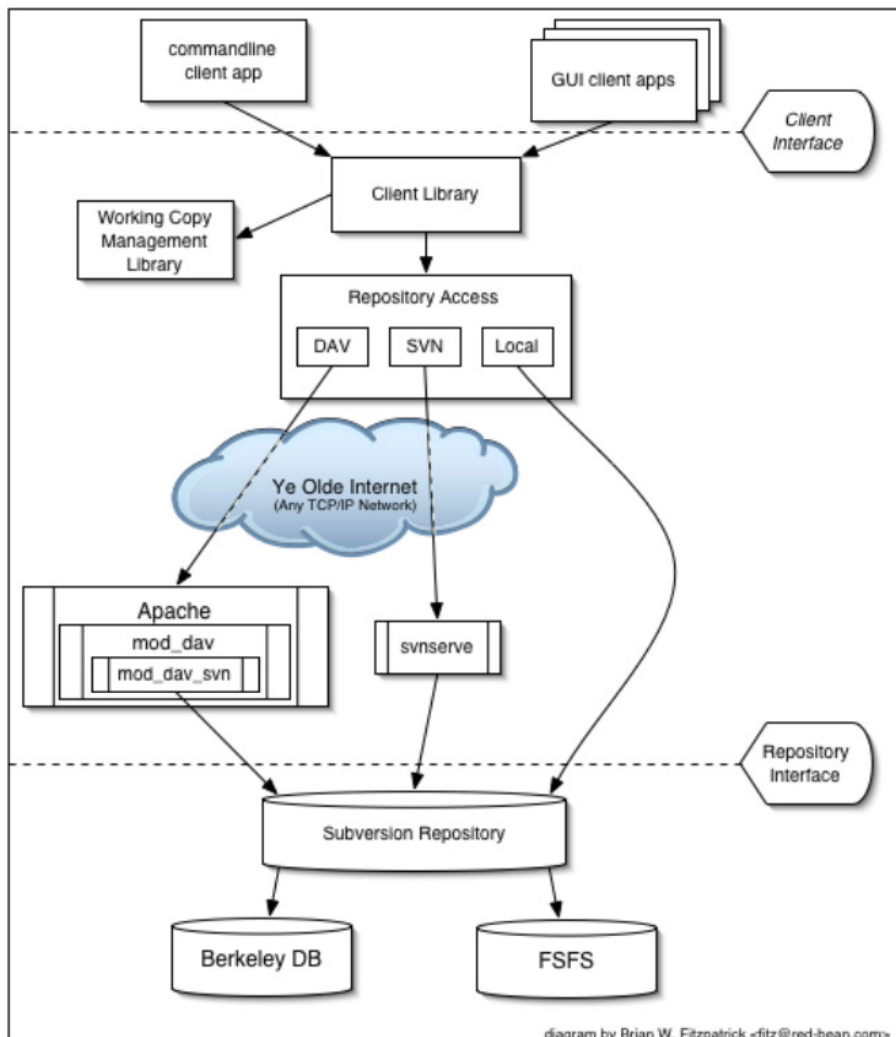
Kehitystiimit toimittavat tuoteversion jokaisessa sprintissä. Jokainen tuoteversio rakentuu kaikkien aikaisempien tuoteversioiden päälle, jotka on testattu läpikotaisiin. Tällä voidaan varmistaa tuoteversioiden toimiminen yhdessä. (21, s. 5–14)

#### **4.2 Versionhallinta**

Versionhallinta helpottaa ohjelmistoprojektien tuotosten hallintaa ja se mahdollistaa ohjelmistokehityksen seurannan ja hallitun kehittämisen. Versionhallinnan avulla voidaan löytää lähdekoodin tai dokumenttien aiemmat versiot ja niiden sisällöt sekä se mahdollistaa myös kaikille kehittäjille pääsyn aina uusia lähdekoodiin tai mahdollisuuden palata aiempiin versioihin. Versionhallinnan avulla on mahdollista saada selville, mikä versio on uusin ja mitä muutoksia siinä on tehty edelliseen verrattuna. Suurempia kokonaisuuksia tai projekteja käsiteltäessä versionhallinnan tärkeys korostuu.

Versiohallinnan voi jakaa neljään eri päätoimenpiteeseen, jotka ovat versiointi, versioiden merkitseminen, versioiden välisten erojen tunnistaminen ja versioiden tallentaminen. Avoimeen lähdekoodiin pohjautuvia versionhallintamenetelmiä ovat esimerkiksi Subversion, Git ja CVS.

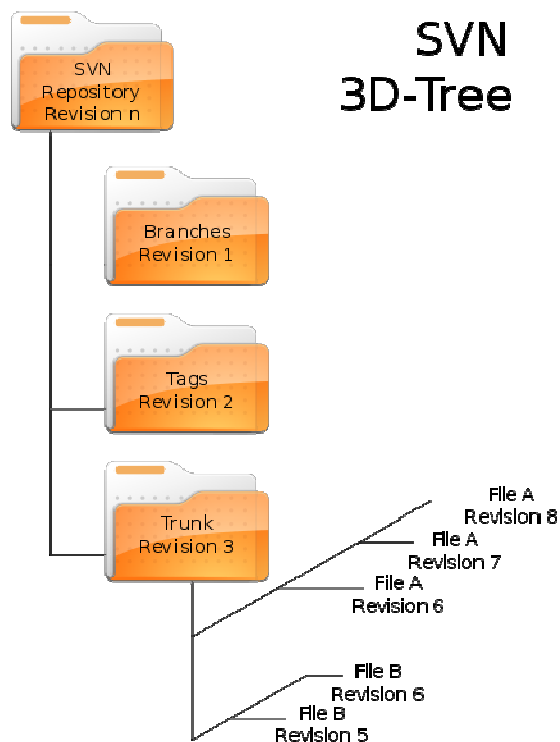
Tässä opinnäytetyössä käytettiin Subversion-versionhallintamenetelmää. Subversion on avoimeen lähdekoodiin pohjautuva versionhallinta järjestelmä (Version Control System, VCS). Subversion hallitsee tiedostot ja kansiot sekä niiden muutokset ajan myötä. Subversion arkkitehtuuri (kuva 11) mahdollistaa uusien ja vanhojen versioiden, sekä niiden sisällön tarkastelun useiden käyttäjien kesken verkon yli.



KUVA 11. Subversionin arkkitehtuuri (22, s. 16)

Subversion kehittämisen on aloittanut CollabNet, Inc. vuonna 2000. Subversionin tarkoitus on lisätä ja korjata CVS:stä puuttuvia yleisesti tarpeellisiksi koettuja ominaisuuksia. Subversion on kehitetty CVS:n pohjalta minkä takia toiminto ja käyttö pohjautuvat monelta osin CVS:n tyyliin. Vuonna 2009 CollabNet aloitti Subversionin integroinnin Apache Software Foundation (ASF) kanssa, joka on maailman suurin avoimen lähdekoodin projektien yhteisö. Subversion liitettiin yhdeksi ASF:n pääprojekteista vuonna 2010 ja se päätettiin uudelleennimetä Apache Subversioniksi. (22, s. 8)

Subversion mahdollistaa yksittäisten muutoksien seuraamista kokonaisuutena, koska commit-toiminto tulee voimaan vasta kun toiminto on kokonaisuudessaan suoritettu. Toimintoa käsitellään yhtenä kokonaisuutena, joka saa oman versionumeronsa (kuva 12).



KUVA 12. SVN:n hierarkia (23)

## 4.3 Työkalut

Tähän lukuun on listattu työkalut joita on käytetty opinnäytetyötä tehdessä. Vaatimuksena oli, että kaikki työkalut perustuvat avoimeen lähdekoodiin.

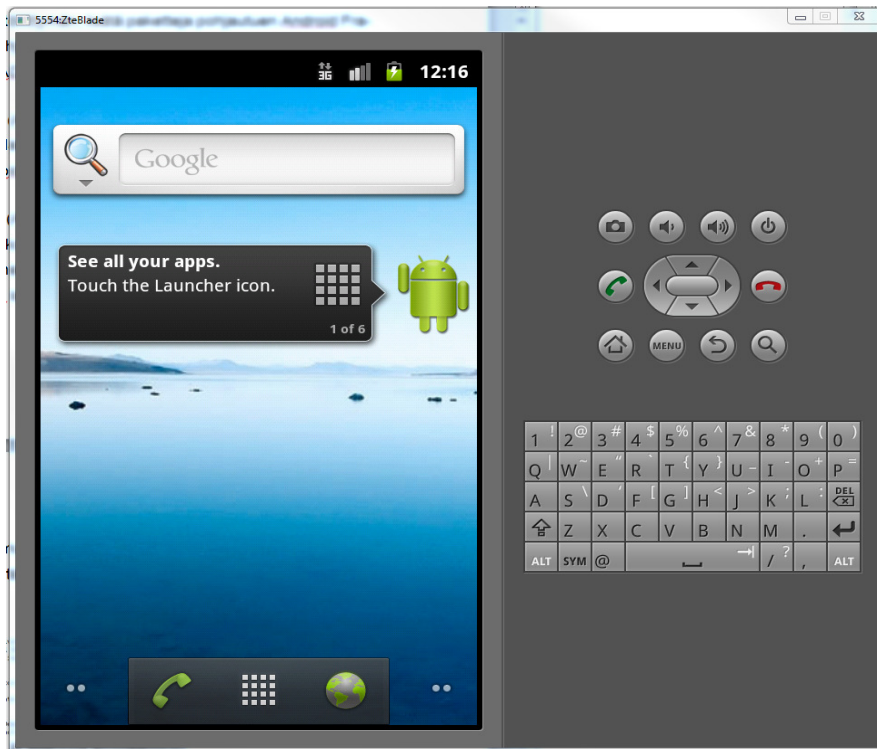
### 4.3.1 Eclipse

Eclipse on alun perin vuonna 1993 IBM:n kehittämä ohjelmointiympäristö. Eclipse julkaistiin avoimen lähdekoodin lisenssillä 2001 ja vuodesta 2004 sen kehityksestä on vastannut Eclipse Foundation (24).

Eclipse-kehitysympäristöstä löytyy useita räätälöityjä jakelupaketteja, kuten Classic, Java, Java EE, JavaScript Web Developers ja C/C++, ja niistä valittiin tähän opinnäytetyöhön käyttöön Eclipse Classic. Eclipse Classiciin asennettiin Googlen ohjeiden mukaan ADT-liitännäinen, jota tarvitaan Android-pohjaiseen ohjelmistokehitykseen. Versiohallintaa varten piti myös asentaa Subversive-liitännäinen.

Android Development Tools (ADT) on Eclipse IDE:n liitännäinen, jonka tarkoituksena on integroida Android-ohjelmien kehitystyökalut Eclipseen. ADT laajentaa Eclipsen toiminnallisuuksia. ADT:llä on mahdollista luoda Android-projekteja sekä ohjelmiston käyttöliittymiä, lisätä paketteja pohjautuen Android Frameworkiin, etsiä ja paikantaa ohjelmointivirheitä ohjelmistoa sekä luoda allekirjoitettu tai allekirjoittamaton APK-tiedosto. ADT-liitännäisen mukana tulee myös AVD ja ADB.

Android Virtual Device (AVD) on emulaattorikonfiguraatio, jolla on mahdollista mallintaa todellinen laite (kuva 13) määrittämällä laitteiston ja ohjelmiston asetukset, joita emuloidaan Android-emulaattorissa. Emulaattorissa on pitkälti samat toiminnot kuin oikeassa Android-älypuhelimessa joten se soveltuu mainiosti ohjelman testaamiseen.



*KUVA 13. Android AVD toiminnassa*

Android Debug Bridge (ADB) on monipuolinen komentorivityökalu, joka mahdollistaa kommunikoinnin emulaattorin tai Android-pohjaisen laitteen välillä. ADB:n avulla on mahdollista siirtää APK-asennustiedosto emulaattoriin tai Android-pohjaiseen laitteeseen (kuva 14). Eclipsessä tämä toiminto on automatisoitu. ADB:n mukana tulevalla logcat-ohjelmalla voidaan myös etsiä ja paikantaa ohjelmistovirheitä sekä kaatumisen mahdollisia syitä. (Kuva 15.)

```

Demo] -----
Demo] Android Launch!
Demo] adb is running normally.
Demo] Performing com.caddieon.demo.DemoActivity activity launch
Demo] Launching a new emulator with Virtual Device 'ZteBlade'
Emulator] emulator: warning: opening audio input failed
Emulator]
Demo] New emulator found: emulator-5554
Demo] Waiting for HOME ('android.process.acore') to be launched...
Demo] HOME is up on device 'emulator-5554'
Demo] Uploading Demo.apk onto device 'emulator-5554'
Demo] Installing Demo.apk...
Demo] Success!
Demo] Starting activity com.caddieon.demo.DemoActivity on device emulator-5554
Demo] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.caddieon.demo/.DemoActivity }
  
```

*KUVA 14. APK-tiedoston siirto ja asennus ADB:n avulla emulaattoriin*

D	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	Shutting down VM
W	05-01 12:09:1...	335	com.caddieon.demo	dalvikvm	threadid=1: thread exiting with uncaught exception (group=0x40015560)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	FATAL EXCEPTION: main
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	java.lang.RuntimeException: Unable to resume activity {com.caddieon.demo...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.performResumeActivity(ActivityThread.java:...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.handleResumeActivity(ActivityThread.java:2...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:1...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.access\$1500(ActivityThread.java:117)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread\$H.handleMessage(ActivityThread.java:931)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.os.Handler.dispatchMessage(Handler.java:99)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.os.Looper.loop(Looper.java:123)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.main(ActivityThread.java:3683)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at java.lang.reflect.Method.invokeNative(Native Method)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at java.lang.reflect.Method.invoke(Method.java:507)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at com.android.internal.os.ZygoteInit\$MethodAndArgsCaller.run(ZygoteInit...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:597)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at dalvik.system.NativeStart.main(Native Method)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	Caused by: java.lang.NullPointerException
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at com.caddieon.demo.GreenActivity.onResume(GreenActivity.java:84)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.Instrumentation.callActivityOnResume(Instrumentation.java:...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.Activity.performResume(Activity.java:3832)
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	at android.app.ActivityThread.performResumeActivity(ActivityThread.java:...
E	05-01 12:09:1...	335	com.caddieon.demo	AndroidRuntime	... 12 more

*KUVA 15. Logcat ilmoittaa toiminnan aikana ilmenneistä virheistä*

Subversive liitännäinen saadaan asennettua suoraan Eclipsestä. Subversiven tarkoituksena on integroida Subversion tuki suoraan Eclipse-kehitysympäristöön jolloin versiohallinta helpottuu. Subversiven avulla projekti voidaan päivittää uusimpaan versioon tai palata vanhempaan versioon sekä projektin muutokset voidaan viedä versiohallintaan nopeasti muutamalla painalluksella.

### 4.3.2 TortoiseSVN

TortoiseSVN on avoimen lähdekoodin Microsoft Windowsin Shell-liitännäinen, joka mahdollistaa Subversion-versionhallinnan tuen suoraan Windows Exploreriin. Koska TortoiseSVN ei ole integraatio millekään tietylle kehitysympäristölle, mahdollistaa se Subversionin käytön kaikkien kehitysympäristöjen sekä tiedostojen kanssa.

### 4.3.3 GIMP

GIMP (GNU Image Manipulation Program) on avoimeen lähdekoodiin perustava monipuolinen kuvankäsittelyohjelma. GIMP on kehitetty toimimaan x11-ikkunointijärjestelmän alla Unixilla, kuten Linux, mutta sama koodi toimii myös Windows- sekä OS X -ympäristöissä.

#### **4.3.4 MediaWiki**

MediaWiki on avoimeen lähdekoodiin perustuva Wiki-ohjelmisto, joka on kirjoitettu käyttäen PHP:ta. Ohjelmistoa käytetään tiedonvälitykseen sekä yhteiskehittelyn alustana. MediaWikissä on mahdollista antaa sekä kaikille käyttäjillä oikeudet lisätä ja poistaa tietoa että antaa oikeudet vain tietyille käyttäjille.



## 5 TYÖN TOTEUTUS

Tämä luku kuvaa Android-pohjaisen ohjelman projektin luomista, hallintaa ja testaamista sekä valmistuneen ohjelman yleisen esittelyn. Esitykset on toteutettu vain yleisellä tasolla ja tarkka osuus on vain tilaajan käytössä.

### 5.1 Projektin aloitus

Alkupalaverissa sovittiin tilaajan kanssa eri vaiheiden tavoiteaikataulut ja sovittiin projektinhallinnan menetelmä. Alkupalaverissa sovittiin myös alustavasti käyttöliittymän toiminnallisuudesta ja ulkoasusta. Myöhemmissä palaverissa sovittiin opinnäytetyöhön liittyvistä muutoksista kuten graafinen ulkoasu, toiminnallisuus, kuvakkeiden sijainnit sekä muut pienet muutokset toiminnallisuudessa. Lähtökohtina olivat asiakkaan toivomukset ja vaatimukset.

Ohjelman käyttöliittymää ei voitu suunnitella suoraan Androidin tyyliohjeen mukaiseksi, koska tilaajalla ei ollut vielä kokemusta Android-käyttöliittymistä. Tämän takia käyttöliittymistä tehtiin vain suuntaa antavat luonnokset ja ohjelmisto toteutettiin niiden pohjalta.

Opinnäytetyötä toteuttaessa käytettiin projektinhallinnassa Scrumia, jossa sprintit oli jaettu kahden viikon sykleihin. Jokaisen sprintin lopuksi pidettiin palaveri, jossa käytiin läpi mitkä osat on saatu tehtyä ja mitä osia tulee seuraavaan sprinttiin. Palaverissa myös suoritettiin julkaisun hyväksyntätestaus Android-pohjaisella ZTE Blade -älypuhelimella.

Ennen ohjelman toteuttamista piti ottaa käyttöön asiakkaan palvelimella versionhallinta sekä asentaa ja ottaa käyttöön palvelimella MediaWiki-ohjelmisto. MediaWikiin luotiin opinnäytetyön ohessa ohjeita esimerkiksi TortoiseSVN:n, Eclipsen ja sen liitännäisten kuten ADT:n ja Subversiven asentamisesta ja käyttöönotosta.

Ohjelman toteuttamiseen valittiin Eclipse ja siihen ADT-liitännäinen, jota Googlekin suosittelee ja tukee. Ohjelma on minulle osittain tuttu, koska olen käyttänyt sitä muutamaan projektiin jo aiemmin. Android-projekteja olin aikaisemmin toteuttanut vain yhden.

## 5.2 Projektin luonti

Projektin luonti tapahtui Eclipsellä. Projekti luotiin Eclipsellä seuraavasti:

- Valitaan Eclipsessä File -> New -> Android Project, jolloin aukeaa projektin luonti ikkuna.
- Auenneessa ikkunassa syötetään projektin nimi ja painetaan Next-painiketta.
- Seuraavasta ikkunasta valitaan kohde, jolle ohjelmisto halutaan kehittää, esimerkiksi Android 2.3.3. Sitten painetaan Next-painiketta.
- Seuraavassa ikkunassa syötetään ohjelman nimi, paketin nimi (esimerkiksi com.projekti.demo) ja Android-alustan vähimmäisvaatimus. Tämän jälkeen painetaan Finish-painiketta.

## 5.3 Projektin versionhallinta

Projektin vieminen versionhallintaan onnistui Subversive-liitännäisen avulla. Tämä edellyttää, että Subversive-liitännäinen on asennettuna jo Eclipseen. Ensimmäiseksi pitää luoda SVN Repositoriesiin kohde. Kohteen luonti onnistuu seuraavasti:

- Eclipsestä valitaan Window -> Open Perspective -> Other...
- Auenneesta ikkunasta valitaan SVN Repository Exploring ja painetaan OK-painiketta.

- SVN Repositories ikkunaan painetaan hiiren oikealla ja valitaan New -> Repository Location...
- Auenneeseen New Repository Location ikkunaan syötetään versionhallinnan tiedot, kuten osoite, käyttäjätunnus ja salasana, ja valitaan Finish.

Kun versionhallinnan kohde on luotu onnistuneesti, voidaan projekti viedä versionhallintaan seuraavasti:

- Auenneeseen New Repository Location ikkunaan syötetään versionhallinnan tiedot, kuten osoite, käyttäjätunnus ja salasana ja valitaan Finish.
- Valitaan projekti Eclipsen Package Explorerista ja painetaan siihen hiiren oikealla painikkeella ja valitaan Team -> Share Project...
- Valitaan SVN ja painetaan Next-painiketta.
- Valitaan Use existing repository locations ja oikea SVN-osoite ja painetaan Finish-painiketta.
- Auenneeseen Commit-valikkoon kirjoitetaan kommentti projektin tiedoista ja syy, miksi se siirretään versionhallintaan. Tarkistetaan, että tiedostot on valittu oikein, ja painetaan OK-painiketta.
- Tämän jälkeen projekti on viety onnistuneesti versionhallintaan.

### **5.3.1 Projektin päivittäminen versionhallintaan**

Kun haluttu muutos on saatu valmiiksi projektiin ja se halutaan tallettaa versionhallintaan asti, voidaan tämä toteuttaa Eclipsellä suoraan seuraavasti:

- Valitaan Eclipsen Package Explorerista haluttu projekti, ja painetaan sitä hiiren oikealla painikkeella ja valitaan Team -> Commit...

- Auenneeseen Commit-valikkoon kirjoitetaan kommentiksi syyt miksi projektia päivitetään. Valikosta myös tarkistetaan, että tiedostot ja niiden tapahtumat ovat oikein jonka jälkeen valitaan OK.
- Jos virheitä ei esiinny, on projekti päivitetty versionhallintaan onnistuneesti.

### 5.3.2 Projektin päivittäminen uusimpaan versioon

Jos halutaan varmistaa, että projekti on varmasti viimeisin versio, voidaan tämä toteuttaa Eclipsellä seuraavasti:

- Valitaan Eclipsen Package Explorerista haluttu projekti ja painetaan sitä hiiren oikealla painikkeella ja valitaan Team -> Update.
- Jos virheitä ei esiinny, on projekti päivitetty onnistuneesti.
- Jos konflikteja esiintyy, päästään niitä tutkimaan valitsemalla Eclipsen Package Explorerista projekti, ja painetaan sitä hiiren oikealla painikkeella ja valitaan Team -> Edit Conflicts.

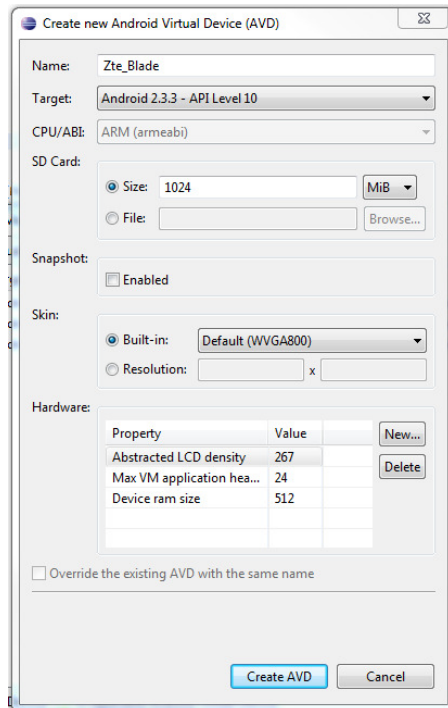
## 5.4 Testaus

Ohjelmaa testattiin alustavasti ensinnä ADT liitännäisen mukana tulevalla AVD:llä (Android Virtual Device), jolle pystyy itse luomaan erilaisia konfiguraatioita laitteesta, jolla ohjelma halutaan saada toimimaan.

AVD:n luonti tapahtuu seuraavasti:

- Valitaan Eclipsestä yläpalkin valinnoista Window ja sen alta AVD Manager.
- Android Virtual Device Managerista valitaan New..., jolloin aukeaa uusi valikko (kuva 16).

- Nyt voidaan syöttää tulevan laitteen nimi ja tiedot, kuten alusta, näyttöresoluutio, näytön tiheys (LCD Density) ja ram-muistin määrä.
- Tämän jälkeen valitaan Create AVD ja AVD:n luonti on valmis.



KUVA 16. Emulaattorin luomisen yhteydessä vaadittavat tiedot

Tässä opinnäytetyössä käytettiin itse luotuja AVD-profiileja. Profiilien ominaisuudet oli asetettu vastaamaan ZTE Blade-, Samsung Galaxy Mini- ja Samsung Galaxy Nexus -puhelimia. Näiden avulla mahdollistettiin ohjelman skaalattavuus valmiiksi useille eri resoluutiolle.

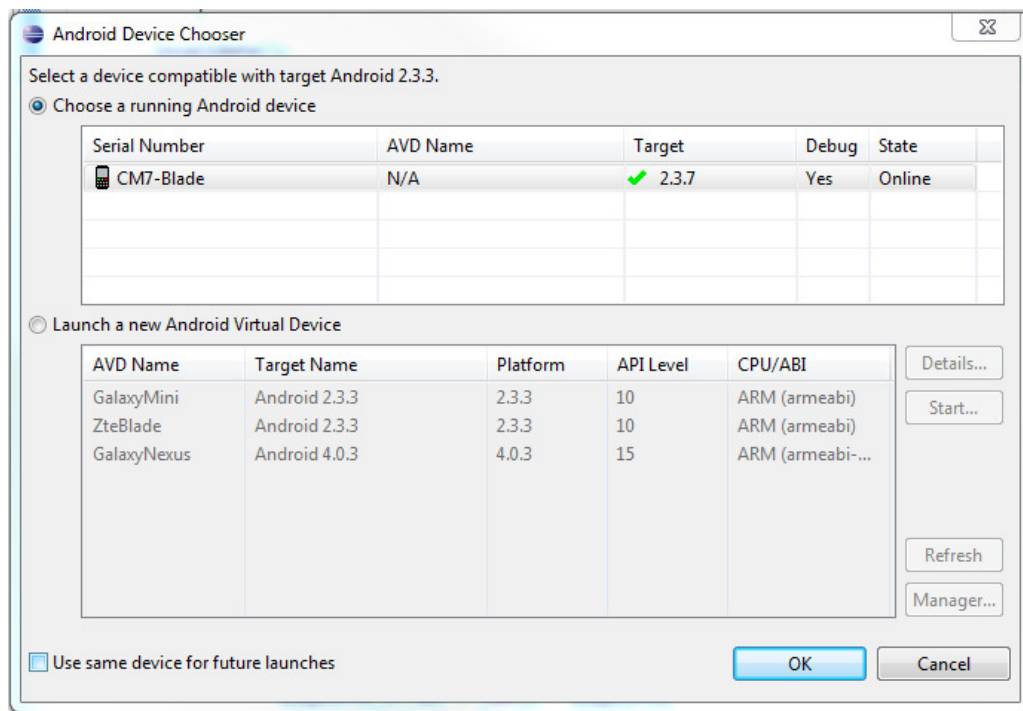
Testipuhelimeksi valittiin ZTE Blade (kuva 17), joka on päivitetty alkuperäisen valmistajan Android-alustan 2.2-versiosta CyanogenMod7 2.3.7-versioksi. Puhelimen asetuksista pitää olla asetettuna Android Debugging päälle, jotta projektin voi viedä suoraan itse puhelimeen sekä suorittaa siellä. Projektin vienti puhelimeen tapahtui Eclipsen kautta:

- Eclipsen Package Explorerista valitaan projekti ja painetaan hiiren oikeaa nappia. Aukeavasta valikosta valitaan Run As -> Android Application.

- Juuri auenneesta Android Device Chooser valikosta (kuva 18) valitaan laite jolle projekti halutaan viedä. Painetaan OK ja odotetaan, että Eclipse vie ohjelmiston ADB:n avulla puhelimeen (kuva 19).



KUVA 17. ZTE Blade -älypuhelin (25)



KUVA 18. Eclipse Device Chooser, josta valitaan laite tai emulaattori, johon APK viedään ja asennetaan

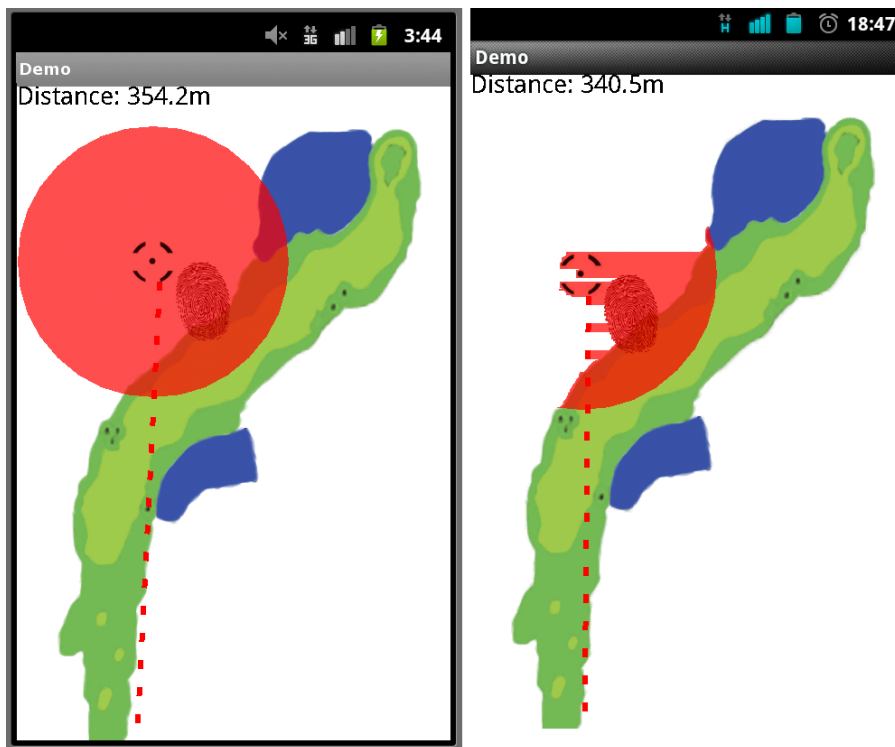
```
Demo] -----
Demo] Android Launch!
Demo] adb is running normally.
Demo] Performing com.caddieon.demo.DemoActivity activity launch
Demo] Uploading Demo.apk onto device 'CM7-Blade'
Demo] Installing Demo.apk...
Demo] Success!
Demo] Starting activity com.caddieon.demo.DemoActivity on device CM7-Blade
Demo] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.caddieon.demo/.DemoActivity }
```

### *KUVA 19. APK-tiedoston onnistunut vienti, asennus ja suorittaminen puhelimessa ADB:n avulla*

Pitkälti ominaisuuksien testaus onnistui emulaattorissa, mutta muutama ominaisuus piti testata suoraan puhelimessa, koska esimerkiksi lyönti tapahtuma tapahtuu lukemalla puhelimen kiihtyvyyssanturia, jonka käyttöä ei löytynyt emulaattorista.

Ohjelma testattiin jokaisen uuden ominaisuuden luonnin ja päivityksen jälkeen suoraan emulaattorissa. Mikäli virheitä ilmeni, ne korjattiin ennen testaamista puhelimessa. Jokaisen sprintin jälkeen julkaistun ohjelman testaus suoritettiin asiakkaan tiloissa ZTE Blade -älypuhelimella.

Ohjelmaa kehitettäessä testauksessa havaittiin useasti, että vaikka ohjelma toimisi oikein emulaattorissa, saattoi ongelmia ilmetä vasta oikeassa laitteessa (kuva 20). Tämän seurauksena ohjelma testattiin jos mahdollista perusteellisesti ensin emulaattorissa ja sitten vasta puhelimessa. Ongelmaksi koettiin myös se, että emulaattorissa kuvakkeet ja teksti saattoivat näyttää selkeiltä ja suurilta, mutta puhelimeen siirrettäessä tekstistä tulikin vaikeaselkoista sekä painikkeet olivat useasti liian pieniä.



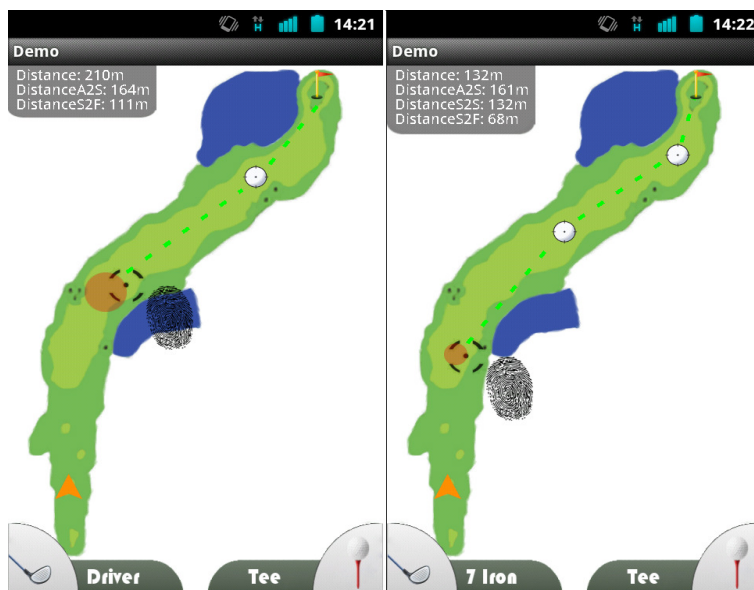
*KUVA 20. Vasemmalla näkymä emulaattorista ja oikealla puhelimesta*



## 6 TULOKSET

Työssä saatiin valmiiksi tilaajan toiveisiin perustuva prototyypiohjelman käyttöliittymä esittelykäyttöä varten Android-pohjaiselle älypuhelimelle. Ohjelman käynnistyessä käyttäjälle avautuu kuva viheriöstä, pelaajan sijainnista ja suositelluista lyöntipaikoista sekä tiedot etäisyyksistä oleellisimpiin kohteisiin. Näitä kohteita ovat matka pelaajasta tähtäyspisteeseen, tähtäyspisteestä suositeltuun seuraavaan pisteeseen (A2S), tarpeen vaatiessa suositellusta pisteestä seuraavaan suositeltuun pisteeseen (S2S) sekä matka viimeisestä suositellusta pisteestä viheriölle (S2F) (kuva 21).

Tähtäyspisteen viereen piirretään punainen kuvitteellinen alue, johon pelaaja todennäköisesti osuu. Tämän alueen koko vaihtelee matkan sekä mailan mukaan (kuva 21). Lyönnin tapahduttua pelaajan uusi sijainti lasketaan punaiselta alueelta satunnaisesti. Ohjelmaan on myös lisätty toiminto jonka tarkoituksena on mahdollistaa epäonnistunut lyönti. Tällöin pelaajan sijainti lasketaan punaiselta alueelta, mutta siihen lisätään satunnaisesti arvottu X:n ja Y:n arvo.



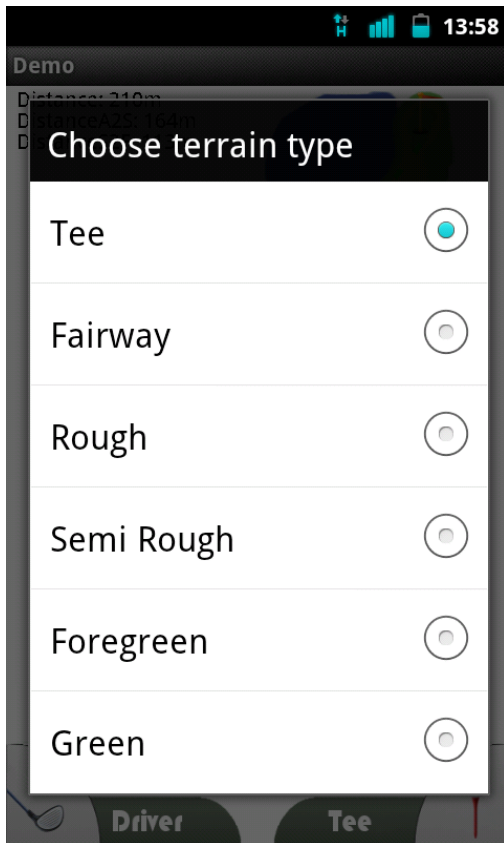
*KUVA 21. Vasemmalla aloitusnäkyminen ohjelmasta ja oikealla näkyminen, jossa käyttäjä on liikuttanut tähtäyspistettä, jolloin lasketaan uudet suositellut jatko-lyöntipaikat.*

Käyttäjän on mahdollista itse siirtää tähtäyspistettä koskettamalla sormenjälkeä ja viemällä tähtäyspiste tämän jälkeen haluttuun kohtaan. Jos tähtäyspiste viedään ohjelman mielestä liian kauas, jonne pelaajan lyönnin ei pitäisi yltää, lopetetaan näyttämästä suositeltua reittiä (kuva 22).



*KUVA 22. Ohjelma ei enää näytä ehdotettua reittiä, jos matka on liian pitkä*

Ohjelma myös tarkkailee pelaajan sijaintia ja ehdottaa lyöntialustaa sen mukaan. Pelaajan on mahdollista myös vaihtaa alustaa painamalla näytön alareunassa näkyvää painiketta, jossa lukee ehdotettu alustan valinta (kuva 23). Ohjelma tarkkailee myös, mihin pelaajan lyönti sijoittuu, ja ilmoittaa, jos pallo lentää veteen (kuva 24). Mikäli lyönti sijoittuu ensimmäisessä aktiviteetissa viheriölle, käynnistetään uusi aktiviteetti, jonne luodaan viheriön näkymä.



*KUVA 23. Ohjelmasta on mahdollista itse valita lyöntialustan tyyppi*



*KUVA 24. Virheilmoitus pallon lentäessä veteen*

Lyönti tapahtuu ohjelmassa niin, että pelaajan heilauttaa puhelinta rivakasti. Ominaisuudessa käytetään hyödyksi puhelimen kiihtyvyyssensoreita (kuva 25). Heilautuksen voimakkuuden ylittäessä tietyn rajan sallitaan lyönti, ja lyöntien välille on asetettu 2,5 sekunnin varoaika. Näillä varmistetaan, että puhelimen pienet heilumiset eivät aiheuta vahinkolyöntejä.

```
private void getAccelerometer(SensorEvent event){
    float[] values = event.values;

    // Movement
    float x = values[0];
    float y = values[1];
    float z = values[2];

    double gforce = Math.sqrt(x * x + y * y + z * z);

    long actualTime = System.currentTimeMillis();

    if (gforce > 25){
        // Waiting time is 2.5s
        if (actualTime - lastUpdate < 2500){
            return;
        }

        lastUpdate = actualTime;

        swing = 1;
    }
}
```

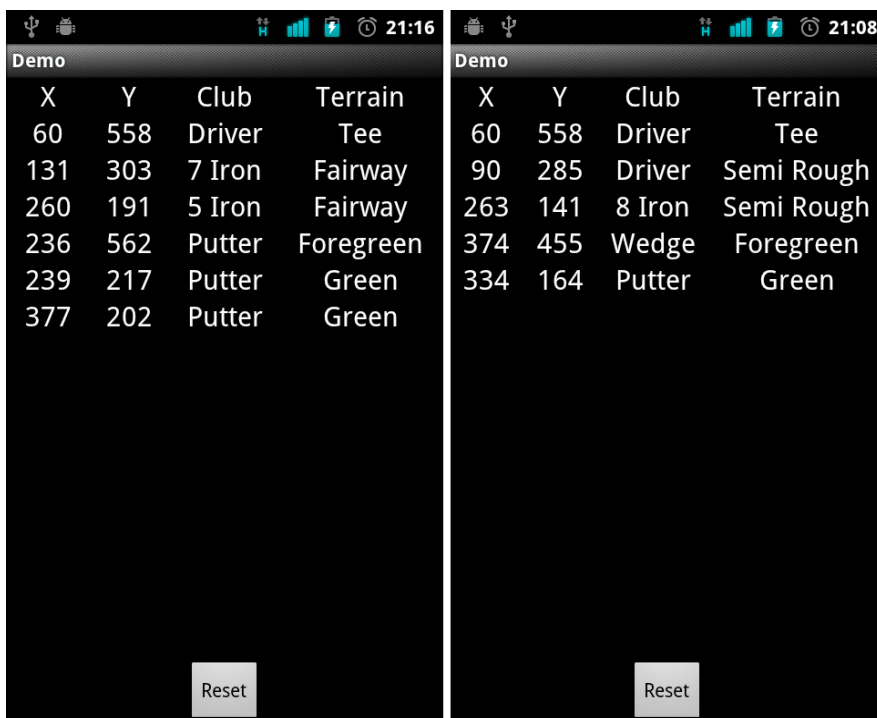
*KUVA 25. Koodiesimerkki, jossa selvitetään puhelimen kiihtyvyyssanturilta heilautuksen voimakkuus.*

Pelaajan saapuessa viheriölle (kuva 26) arvotaan pelaajan sijainti sekä merkkataan pelaajan sijainnin perusteella alustan tyyppi ja mailavalinta. Viheriöllä ei enää näytetä tähtäyspistettä vaan oletetaan aina, että pelaaja tähtää lipua kohden. Kunkin lyönnin yhteydessä pelaajan paikka arvotaan ja tarkistetaan, onko pelaaja saanut pallon reikään. Todennäköisyys sille saako pelaaja pallon reikään, vaihtelee etäisyyden mukaan.



KUVA 26. Viheriön näkymä

Kun pallo saadaan lopulta reikään, aloittaa ohjelma uuden aktiviteetin. Aktiviteetissa luodaan näkymä, jossa on tiedot pelaajan lyöntikoordinaateista sekä käytetyistä mailoista ja alustan tyypistä (kuva 27).



KUVA 27. Ohjelman viimeinen aktiviteetti

## 7 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli toteuttaa tilaajan toiveisiin perustuva prototyyppiohjelman käyttöliittymä esittelykäyttöä varten Android-pohjaiseen älypuhelimeen. Käyttöliittymän piti toimia ZTE Blade -älypuhelimessa. Työn tuloksena syntyi asiakkaan toiveisiin perustuva ohjelma. Ohjelmassa on valmiudet skaalautuvuuteen erikokoisten näyttöjen kanssa, mutta niiden toteutukset eivät olleet prioriteettilistalla.

Aikataulun rajallisuuden vuoksi päätettiin yhdessä tilaajan kanssa, että opinnäytetyöhön otettaisiin vain ”Pelaaja valmistautuu lyöntiin”-skenaario, joka sisältää yhden kentän pelaamisen virtuaalisesti. Alkuvalikot ja muut osiot voidaan jättää tuleviin projekteihin.

Opinnäytetyöstä oli minulle hyötyä siten, että sain kokemusta ohjelmoimisesta Android-pohjaisille laitteistoille sekä Eclipse-kehitysympäristön käytöstä. Sain myös kokemusta ohjelmistoprojekteihin liittyvistä tärkeistä elementeistä, kuten projektinhallintamenetelmät sekä versionhallinta. Näiden kokemusten hyödyntäminen on varmasti tarpeen tulevaisuudessa.

Jatkokehityksessä voidaan suunnitella käyttöliittymän pohjalta toimiva ohjelmisto, johon voidaan toteuttaa myös kaikki loput toiminnallisuudet. Tulevaisuudessa Android-pohjaisen ohjelmiston kehitys onnistuu Tuomo Lallilta sekä hänen tulevan tiiminsä jäseniltä Eclipsellä ja sen liitännäisellä. Näiden käyttöönotosta on tehty kuvalliset ohjeet, joissa kuvataan kohta kohdalta, kuinka ohjelmistot asennetaan ja kuinka ne otetaan käyttöön.

## LÄHTEET

1. Yritystakomo. 2012. Saatavissa: [www.yritystakomo.fi/takomosta/](http://www.yritystakomo.fi/takomosta/). Hakupäivä 29.3.2012.
2. Android (Operating System). 2012. Saatavissa: [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)). Hakupäivä 29.3.2012.
3. Virki, T – Carew, S. 2011. Google topples Nokia from smartphones top spot. Saatavissa: <http://uk.reuters.com/article/2011/01/31/oukin-uk-google-nokia-idUKTRE70U1YT20110131>. Hakupäivä 24.4.2012.
4. Pettey, C. 2011. Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent. Saatavissa: <http://www.gartner.com/it/page.jsp?id=1848514>. Hakupäivä 24.4.2012.
5. Mobiputing. 2011. Saatavissa: [http://mobiputing.com/wp-content/uploads/2011/11/ics\\_17.jpg](http://mobiputing.com/wp-content/uploads/2011/11/ics_17.jpg). Hakupäivä 24.4.2012
6. What is Android. 2012. Saatavissa: <http://developer.android.com/guide/basics/what-is-android.html>. Hakupäivä 24.4.2012.
7. Android System Architechure. 2012. Saatavissa: <http://developer.android.com/images/system-architecture.jpg>. Hakupäivä 24.4.2012.
8. Android Activities. 2012. Saatavissa: <http://developer.android.com/guide/topics/fundamentals/activities.html>. Hakupäivä 30.4.2012.

9. Android Activity Lifecycle. 2012. Saatavissa: [http://developer.android.com/images/activity\\_lifecycle.png](http://developer.android.com/images/activity_lifecycle.png). Hakupäivä 30.4.2012
10. AndroidManifest. 2012. Saatavissa: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>. Hakupäivä 30.4.2012.
11. Google Play for Developers. 2012. Saatavissa: <https://support.google.com/googleplay/android-developer/bin/topic.py?hl=en&topic=15867>. Hakupäivä 24.4.2012.
12. Nazeer, U. 2012. How Many Android Apps are There in Google Play. Saatavissa: <http://www.geeknaut.com/how-many-android-apps-are-there-10197157.html>. Hakupäivä 6.5.2012.
13. Industry Leaders Announce Open Platform for Mobile Devices. 2007. Saatavissa: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html). Hakupäivä 24.4.2012.
14. OHA FAQ. 2012. Saatavissa: [http://www.openhandsetalliance.com/oha\\_faq.html](http://www.openhandsetalliance.com/oha_faq.html). Hakupäivä 24.4.2012
15. Android topples Symbian as the world's most popular smartphone operating system. 2011. <http://www.dailymail.co.uk/sciencetech/article-1352426/>. Hakupäivä 6.5.2012.
16. Mikä on Android. 2012. Saatavissa: <http://blog.androidsuomi.fi/mika-on-android/>. Hakupäivä 6.5.2012.
17. Tofel, K. 2011. Android vs. iPad: the tablet sales figures that matter. Saatavissa: <http://gigaom.com/mobile/android-vs-ipad-the-tablet-sales-figures-that-matter/>. Hakupäivä 24.4.2012.



18. Taulutietokoneet. 2011. Saatavissa:  
<http://www.matkapuhelimet.net/taulutietokoneet/>. Hakupäivä 24.4.2012.
19. Lindström, J. 2011. Scrum. Saatavissa:  
<http://reaktor.fi/osaaminen/scrum/>. Hakupäivä 24.4.2012.
20. Scrum Process. 2011. Saatavissa:  
[http://reaktor.fi/assets/images/scrum\\_process.png](http://reaktor.fi/assets/images/scrum_process.png). Hakupäivä 24.4.2012.
21. Schwaber, K – Sutherland, J. 2011. The Scrum Guide. Suom. Lekman, L. Heljo, S. Saatavilla:  
<http://www.scrum.org/storage/scrumguides/Scrum%20Guide%20-%20FI.pdf>.  
Hakupäivä 28.4.2012
22. Collins-Sussman, B – Fitzpatrick, B – Pilato, C. 2011. Version Control with Subversion. Saatavissa: <http://svnbook.red-bean.com/en/1.7/svn-book.pdf>. Hakupäivä 29.4.2012.
23. SVN 3D-tree. 2008. Saatavissa:  
[http://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Svn\\_3D-tree.svg/250px-Svn\\_3D-tree.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/0/03/Svn_3D-tree.svg/250px-Svn_3D-tree.svg.png). Hakupäivä 29.4.2012.
24. Eclipse (IDE). 2012. Saatavissa: [http://fi.wikipedia.org/wiki/Eclipse\\_\(IDE\)](http://fi.wikipedia.org/wiki/Eclipse_(IDE)).  
Hakupäivä 29.4.2012.
25. ZTE Blade. 2011. Saatavissa:  
[http://www.elisashopit.fi/pr\\_img/big/4ceb910397382.png](http://www.elisashopit.fi/pr_img/big/4ceb910397382.png). Hakupäivä 6.5.2012.