

MOBIILISOVELLUSTEN KEHITTÄMINEN

Ilkka Juola

Opinnäytetyö
Huhtikuu 2011

Tietojenkäsittely
Luonnontieteiden ala





Tekijä(t) JUOLA, Ilkka	Julkaisun laji Opinnäytetyö	Päivämäärä 14.03.2011
	Sivumäärä 48	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkojulkaisulupa myönnetty (X)
Työn nimi MOBIILISOVELLUSTEN KEHITTÄMINEN		
Koulutusohjelma Tietojenkäsittely		
Työn ohjaaja(t) TUIKKA, Tommi		
Toimeksiantaja(t) Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma		
Tiivistelmä <p>Työn tilasi Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma. Tutkimuksen tarkoituksena oli selvittää mobiilisovelluksen elinkaarta aina alustan valinnasta sen julkaisuun ja markkinointiin. Lisäksi haluttiin tutkia sovellusten kaupallistamista tarkemmin. Näkökulmaksi valittiin yksityiskäyttäjät.</p> <p>Opinnäytetyön tutkimustavaksi valittiin laadullinen tutkimus. Teoriaosuus kasattiin kirjallisista lähteistä. Sovellusten kehitystyökaluja (mm. Xcodea) ja ohjelmointia kokeiltiin omatoimisesti. Sovelluskauppojen analyysit luotiin kolmannen osapuolen lähteiden pohjalta.</p> <p>Tutkimuksista kävi ilmi, että mobiilisovellusten kehitystyössä on ohjelmoinnin lisäksi otettava huomioon paljon muita teknisiä seikkoja. Älypuhelinien ja muiden mobiililaitteiden käyttöjärjestelmät päivittyvät tiheästi, ja tämä vaikuttaa oleellisesti kehitysprosessiin. Sovelluskaupat ovat kasvaneet miljardiluokan liiketoiminnaksi. Niitä analysoidessa kuitenkin huomattiin, että ansaintamahdollisuudet ovat lopulta keskimäärin pieniä. Tutkimuksesta kävi myös ilmi, että mobiilisovellusta markkinoitaessa jatkuva päivittäminen ja ahkera asiakaspalvelu näyttelevät suurta osaa.</p> <p>Työn tuloksia on tarkoitus käyttää opetustilanteiden tukena. Etenkin liiketoiminnallista näkökulmaa voidaan tuoda esille perinteisen ohjelmoinnin opetuksen lisäksi.</p>		
Avainsanat (asiasanat) Mobiilisovellus, mobiilikäyttöjärjestelmä, Android Market, App Store		
Muut tiedot		



Author(s) JUOLA, Ilkka	Type of publication Bachelor's Thesis	Date 14.03.2011
	Pages 48	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title MOBILE APPLICATION DEVELOPMENT		
Degree Programme Business Information Systems		
Tutor(s) TUIKKA, Tommi		
Assigned by JAMK University of Applied Sciences		
Abstract <p>The research was carried out for the study program Business Information Systems of JAMK University of Applied Sciences. The purpose of the thesis was to find out what goes into making a mobile application from choosing a mobile operating system to commercializing, releasing and marketing. An independent developer's point of view was chosen.</p> <p>The primary research method was to collect information from books, newspapers and other articles. A small experiment was used to test development tools and programming. For the purposes of analyzing mobile application distribution platforms, 3rd party numbers and data were used.</p> <p>As results of this thesis it was found out that besides programming, a lot of other technical knowledge is required to develop mobile applications. Mobile software is constantly moving forward and thus can cause problems during designing and building applications. It was also found that, while mobile application business is booming, most apps do not really get much attention. Additionally, it seems that making it as a professional in the business is fairly tough and requires a great deal of luck.</p> <p>The results will be used to provide additional knowledge for the purposes of teaching mobile application development at JAMK University of Applied Sciences.</p>		
Keywords Mobile application, mobile operating system, Apple App Store, Android Market		
Miscellaneous		

SISÄLTÖ

1	JOHDANTO	3
2	TUTKIMUSASETELMA.....	4
2.1	Tutkimuksen taustaa, tavoitteet ja rajaukset.....	4
2.2	Tutkimusmenetelmät	5
2.3	Tutkimuskysymykset	5
3	ÄLYPUHELIN JA MOBIILISOVELLUKSET	6
3.1	Mobiilikäyttöjärjestelmät	7
3.2	Mobiilisovellus	10
3.3	Sovellusten jakelukanavat.....	11
4	MOBIILIALUSTAN VALITSEMINEEN	13
4.1	Symbian, QT ja niiden tulevaisuus	13
4.2	Androidille kehittäminen pähkinänkuoressa	15
4.2.1	Koodiesimerkki	16
4.2.2	App Inventor	17
4.2.3	Haasteet	18
4.3	iOS:ille kehittäminen pähkinänkuoressa.....	19
4.3.1	Koodiesimerkki	20
4.3.2	iOS-laitteet.....	21
4.4	Selainpohjaiset sovellukset	23
4.5	Yhteenveto	24
5	SOVELLUKSEN JULKAISU	25
5.1	Allekirjoitus	25
5.2	Ohjelmakuvaus.....	25
5.3	Tarkastusprosessi	27
6	SOVELLUSKAUPPOJEN ANSAINTAMALLIT	29
6.1	Tapaus Angry Birds.....	29
6.2	Sovelluksen myynti.....	30
	Esimerkkilaskelmia.....	30

Hinnoittelu	33
6.3 Freemium	34
6.4 Mainokset.....	36
6.5 Kumppanuusmarkkinointi	38
6.6 Tilausmalli	39
7 JULKAISUN JÄLKEEN.....	40
7.1 Analyysityökalut.....	41
7.2 WWW-sivut, sosiaalinen media	41
7.3 Päivitykset	42
7.4 Asiakaspalaute	43
7.5 Ilmaisia promootiokeinoja	44
8 POHDINTA.....	45
LÄHTEET	47
KUVIOT	
KUVIO 1. Mobiilikäyttöjärjestelmien markkinaosuudet vuonna 2009 ja 2010 ...	8
KUVIO 2. Hello World -sovellus Androidissa	17
KUVIO 3. HelloWorldAppDelegate.m -tiedosto.	21
KUVIO 4. MyView.h -tiedosto.	21
KUVIO 5. Kuinka iPhoneen käyttäjät löytävät uusia sovelluksia?.....	26
KUVIO 6. Android Marketin latausten jakautuminen.	32
KUVIO 7. Lisäsisällön myynnin kehittyminen mainostuloihin verrattuna.....	35
KUVIO 8. Mainoksia Angry Birds pelin Android-versiossa.....	36
TAULUKOT	
TAULUKKO 1. Android -versioiden jakautuminen.	19
TAULUKKO 2. Käyttöjärjestelmäversioiden saatavuus iOS -laitteissa.	22
TAULUKKO 3. Yhteenveto mobiilialustoista.....	24

1 JOHDANTO

Älypuhelinien ja muiden mobiililaitteiden suosio on kasvanut viime vuosien aikana paljon. Ne ovat levinneet yritys- ja harrastekäyttäjien käsistä koko kansan käyttöön. Älypuhelimet ovat taskukokoisia tietokoneita ja ne toivat mukanaan liikkuvan Internetin.

Älypuhelimet toivat mukanaan myös kolmannen osapuolen mobiilisovellukset ja niiden jakamiseen keskittyvät sovelluskaupat, joiden räjähdysmäinen suosio on ollut vielä vaikuttavampaa kuin itse laitteiden. Sovelluksia on ollut mahdollista ostaa ja asentaa puhelimiin jo vuosituhatien alkupuolelta lähtien, mutta niiden tärkeys on noussut esille vasta viime vuosina. Niiden suosio on kasvanut niin suureksi, että niiden ympärille on kasvanut miljardiluokan liiketoiminta.

Tässä työssä tutkitaan mobiilisovelluksen elinkaarta aina alustan valinnasta sen julkaisuun ja markkinointiin. Näkökulmana on yksityiskäyttäjä ja oletuksena on, että sovelluksen idea on valmiina. Tutkimuksessa pyrittiin selvittämään mitä tekijöitä kannattaa ottaa huomioon ennen ohjelmointiin ryhtymistä, kuinka sovelluskaupat toimivat, mitkä ovat niiden ansaintamallit ja millaista markkinointia voidaan suorittaa kengännauhabudjetilla.

2 TUTKIMUSASETELMA

2.1 Tutkimuksen taustaa, tavoitteet ja rajaukset

Opinnäytetyön taustalla on ajatus siitä, että kuka tahansa voi saada idean, josta tulisi hänen mielestään hyvä ja/tai hyödyllinen mobiilisovellus. Idea voi olla täysin omaperäinen tai vaikkapa paranneltu versio jo olemassa olevasta konseptista tai sovelluksesta. Markkinoita tuntuu olevan, mutta miten oman tuotteensa saa sinne näkyviin? Mitkä ovat askeleet idean kehittämisestä julkaisuun? Mitä voi odottaa julkaisun jälkeen? Sovellusten ympärille on noussut miljardiluokan liiketoiminta. Rahaa tekevät niin yrityksen kuin itsenäiset kehittäjätkin. Mediassa yksityisten ihmisten kohdalla tietysti korostuvat menestystarinat, jossa kehittäjä on noussut vastoin kaikkia todennäköisyyksiä ”ryyysistä rikkauksiin”.

Halusin tutkia tarkemmin kyseistä liiketoimintaa ja selvittää, kuinka kannattavaa se onkaan kehittäjille. Sovelluskaupat pursuavat tarjontaa, joten kuinka taattua siellä menestyminen lopulta on? Ideaalisin tilanne olisi luoda itse sovellus joko omiin tai toimeksiantajan tarpeisiin, julkaista se ja tutkia sen menestymistä. Omalla kohdalla tämä ei ollut mahdollista, koska toteutuskelpoinen idea puuttui ja käytössä oleva opintoaika oli rajallinen. Tiesin, että koulutusohjelmassani on mahdollista opiskella mobiilisovellusten ohjelmointia ja kävi ilmi, että sitä ollaan lisäämässä opintosuunnitelmaan tulevaisuudessa. Yhteistyötä päätettiin tehdä ja näin toimeksiantajana on Jyväskylän ammattikorkeakoulun tietojenkäsittelyn koulutusohjelma.

Tämän työn tavoitteena on luoda informatiivinen yleiskatsaus mobiilisovellusten kehittämistyökaluista, jakelukanavista, ansaintamalleista ja ruohonjuuritason markkinoinnista. Kaikissa kohdissa pyritään ottamaan huomioon yksityiskäyttäjän näkökulma ja niin sanottu kengännauhahudjetti. Työssä käsitellään ohjelmointia, mutta pääpaino ei ole ohjelmointioppaan tuottamisessa. Tarkoituksena on tuottaa tietoa, joka tukisi mobiilisovellusten kehittämistä opetustilanteessa.

Sovelluksen kehittämistä ei rajata suosituimpiin puhelimiin, koska eri valmistajat käyttävät eri käyttöjärjestelmiä, jolloin valmistajat ja puhelinmallit ovat tavallaan irrelevantteja. Sen sijaan rajaus tehdään kolmeen suosituimpaan mobiilialustaan, joita ovat Symbian (Qt), iOS ja Android. Jakelukanavien eli sovelluskauppojen käsittely rajataan kahteen markkinajohtajaan eli App Storeen ja Android Marketiin. Markkinointi rajataan konsepteihin, joita voidaan yleisesti ottaen pitää ilmaisina.

2.2 Tutkimusmenetelmät

Teoriapohja kasataan kirjallisista lähteistä. Ohjelmointia ja kehitystyökaluja käsitellään pienen omakokeilun perusteella. Sovelluskauppojen markkina- ja ansainta-analyysit perustuvat kolmannen osapuolen tilastoihin. Näitä tilastoja yhdistelemällä ja analysoimalla voidaan tehdä erilaisia johtopäätöksiä ja vastata tutkimuskysymyksiin.

2.3 Tutkimuskysymykset

1. Mitä vaaditaan mobiilisovelluksen toteuttamiseen erityyppisille alustoille ja mitä kehitysvaiheessa tulisi ottaa huomioon?
2. Kuinka mobiilisovellusten yleisimmät jakelukanavat toimivat ja mitkä ovat niiden ansaintamallit?
3. Kuinka tuottoisaa mobiilisovellusten tekeminen keskimäärin on ja miten pienellä markkinoinnilla voidaan parantaa menestymismahdollisuuksia?

3 ÄLYPUHELIN JA MOBIILISOVELLUKSET

Älypuhelimien määrittelyyn ei ole olemassa niin sanotusti oikeaa tapaa. Hyvin karkeasti voidaan sanoa, että puhelin on älykäs, kun se kykenee perinteisten matkapuhelinominaisuuksien lisäksi suoriutumaan kämmentietokoneen tehtävistä. Olennaista ei niinkään ole se, mikä yritys puhelimen on valmistanut, vaan että puhelin sisältää ”älykkään” käyttöjärjestelmän, johon kolmannen osapuolen kehittäjät voivat luoda sovelluksia ja muuta sisältöä. IBM:n valmistamaa Simonia pidetään yleisesti ottaen ensimmäisenä älypuhelimien esisänä. Se tuli markkinoille vuonna 1993 ja ominaisuuksia olivat mm. kosketusnäyttö, hakulaite, kalenteri, sähköposti ja laskin. Laite oli isokokoinen ja kärsi teknisistä vioista, minkä vuoksi se vedettiin markkinoilta jo seuraavana vuonna. (Reed, 2010.)

Nokian Communicator-sarjan puhelimet olivat ensimmäisiä nykypäivän älypuhelimia muistuttavia laitteita. Erityisen huomionarvoinen malli on vuonna 2001 julkaistu 9210, jossa oli värinäyttö ja ensimmäistä kertaa käyttöjärjestelmänä kehittäjille avoin Symbian OS sekä QWERTY-näppäimistö. Laitteen ominaisuuksia olivat mm. sähköposti, faksi, Internet-selain, WAP ja Office-paketti, joka sisälsi tekstinkäsittely- ja taulukkolaskentaohjelmat. Uudempiin Communicator-laitteisiin päivitettiin mukaan WLAN, kamera, EDGE, Bluetooth ja MP3-soitin. Muita 2000-luvun alkupuolen älypuhelinvalmistajia olivat Ericsson, Handspring (nykyään Palm) ja BlackBerry. Symbian OS:n lisäksi muita laajemmalle levinneitä käyttöjärjestelmiä olivat Microsoftin Windows CE ja Palm OS. (Reed B, 2010.)

Modernin älypuhelimien valloitus kuitenkin alkoi vuonna 2007, kun Apple lanseerasi oman puhelinmallinsa, iPhoneen. Se oli puhelimen ja heidän suosittu mediasoitin, iPodin, yhdistelmä. Se oli myös ensimmäisten puhelimiensa joukossa, jonka käyttö perustui lähes täysin kosketusnäyttöön. iPhone oli myös ensimmäinen puhelin markkinoilla, jonka näyttö kykeni monikosketukseen eli reagoimaan kahteen tai kolmeen yhtäaikaan sormen painallukseen. Puhelin sisälsi hiotun käyttöliittymän lisäksi myös edistyksellisen web-selaimen, jonka ansiosta Internet-sivustojen selaaminen oli helppoa ja koke-

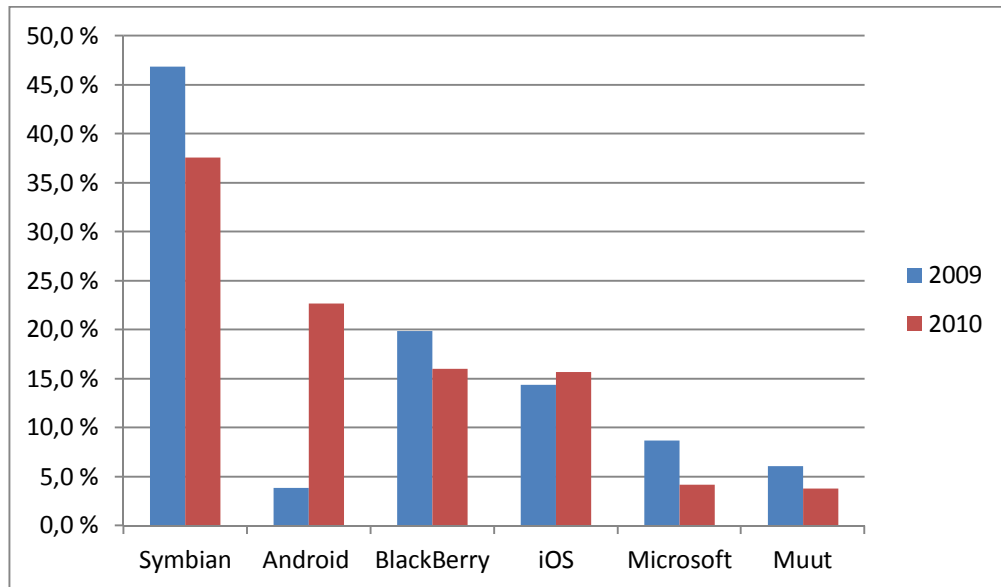
mus verrattavissa tietokoneen selaimen käyttöön. Seuraavana vuonna Apple julkaisi toisen sukupolven iPhoneen, joka sisälsi käytännössä kaikki ominaisuudet, joita tänä päivänä mielletään älypuhelimien sisältävän. Samaan aikaan lanseerattiin myös App Store, josta laitteeseen oli mahdollista ladata ulkopuolisten tekemiä sovelluksia ja muuta sisältöä. (Schedeen 2010.)

Tänä päivänä eri puhelinvalmistajia on kymmeniä ja eri malleja satoja. Modernin älypuhelimien ominaisuuksia ovat tyypillisesti:

- iso kosketusnäyttö
- fyysinen QWERTY-näppäimistö
- WLAN, 3G, GPRS, EDGE, Bluetooth, GPS
- web-selain
- sähköposti ja pikaviestinmahdollisuudet
- sosiaalinen media integroituna
- mediasoitin joka kykenee toistamaan kuvaa, ääntä ja videota
- kamera joka kykenee myös videokuvaukseen
- mahdollisuus asentaa kolmannen osapuolen sovelluksia
- tuki muistikortille.

3.1 Mobiilikäyttöjärjestelmät

Mobiilikäyttöjärjestelmän tehtävä on sama kuin perinteisen tietokoneen käyttöjärjestelmän, eli laitteen aivoina toimiminen. Lisäksi niiden käyttö muistuttaa tietokoneiden käyttöjärjestelmiä, mutta ominaisuudet ovat rajallisempia ja hieman yksinkertaisempia. Älypuhelimien lisäksi mobiilikäyttöjärjestelmiä käytetään muissa mobiililaitteissa, kuten taulu- ja kämmentietokoneissa. IT-alan tutkimus- ja konsultointiyritys Gartnerin mukaan tällä hetkellä suosituimmat järjestelmät ovat Symbian OS, Android, BlackBerry ja iOS. Gartner ennustaa, että tulevaisuudessa Android tulee kasvattamaan ja Symbian menettämään eniten markkinaosuutta. Mainitut järjestelmät tulevat todennäköisesti olemaan suurimmat tekijät vielä pitkään, koska uusia kilpailijoita on vähän ja mobiili-markkinoille on vaikea saada jalansijaa. (Gartner 2011.)



KUVIO 1. Mobiilikäyttöjärjestelmien markkinaosuudet vuonna 2009 ja 2010. (Gartner 2011)

Android

Android -käyttöjärjestelmän tarina on suhteellisen tuore verrattuna muihin kilpailijoihin. Vuonna 2005 Google osti pienen starttiyrityksen Android Inc:n. Tämän jälkeen Google alkoi kehittää omaa Linuxiin perustuvaa mobiilikäyttöjärjestelmää älypuhelimia sekä taulu- ja kämmentietokoneita varten. Vuonna 2007 Google ja 78 muuta yritystä perustivat Open Handset Alliancen (OHA), yritysliittoutuman, jonka tarkoituksena on luoda avoimia standardeja mobiililaitteille. Mukana olevia yrityksiä ovat mm: Dell, Intel, Motorola, HTC, Samsung ja Nvidia. Samalla Android myös julkaistiin virallisesti. Android perustuu avoimeen lähdekoodiin ja on avoimen lisenssin alla ilmainen. Tämä tarkoittaa sitä, että laitevalmistajat voivat käyttää Androidia ilmaiseksi. He voivat myös luoda omaa ilmaista tai maksullista sisältöä Androidin päälle. Valmistajat eroavat toisistaan mm. muokatuilla käyttöliittymillä. Ensimmäinen Androidia hyödyntävä älypuhelin markkinoilla oli HTC Dream, joka ilmestyi vuonna 2008. Androidin suosio puhelimissa on kasvanut lyhyessä ajassa erittäin suureksi. Gartnerin mukaan laitemyynti oli vuonna 2010 kasvussa 888,8 % edellisvuoteen verrattuna. (Beavis 2008.)

iOS

Applen iPhone OS, (myöhemmin pelkkä iOS), kehitettiin alkujaan iPhoneen käyttöjärjestelmäksi ja julkaistiin samaan aikaan puhelimen esittelyn yhteydessä vuonna 2007. IOS perustuu kevyesti Applen Mac OS X:ään, jota yhtiö käyttää tietokoneille. Järjestelmä on suljettu ja sitä ei voi käyttää kolmannen osapuolen laitteissa. Ensimmäinen versio oli vielä suljettu kehittäjiltä, mutta vuonna 2008 App Storen avaamisen yhteydessä julkaistiin uusi versio, joka mahdollisti sovellusten kehittämisen ulkopuolisille. Sitten iOS:in käyttö on levinnyt myös muualle Applen tuoteperheeseen. Käyttöjärjestelmää hyödyntäviä laitteita iPhoneen lisäksi ovat mediasoitin iPod Touch, taulutietokone iPad ja Apple TV. Kaiken kaikkiaan iOS:in markkinaosuus vuonna 2010 oli 15,7 %. (Schedeen 2010.)

Symbian

Symbianin juuret ulottuvat kaikista mobiilikäyttöjärjestelmistä pisimmälle. Sen tarina alkoi alun perin jo vuonna 1980 Psion-ohjelmistotalosta. Psion valmisti alun perin pelejä ja ohjelmistoja, mutta siirtyi kämmentietokoneiden kautta graafisen käyttöjärjestelmän luomiseen. Merkittävin versio siitä oli vuonna 1997 julkaistu EPOC32. Seuraavana vuonna matkapuhelinvalmistajat Nokia, Motorola ja Ericsson loivat yhteisyrityksen nimeltä Symbian, ostivat kyseisen käyttöjärjestelmän ja nimesivät sen SymbianOS:ksi. Ensimmäiset Symbianilla varustetut puhelimet tulivat markkinoille jo vuosituhannen alussa. Tästä alkoi Symbianin tähän päivään saakka kestänyt markkinajohto. Nokia on vastannut tuotekehityksestä yksin vuodesta 2008 lähtien, kun se osti omistuksen täysin itselleen muilta valmistajilta. (The SymbianInvestor n.d.)

Nokia ja Microsoft julkistivat 11.2.2011 tiedon suuresta yhteistyökumppanuudesta. Nokia siirtyisi käyttämään älypuhelimissaan Microsoftin kehittämää Windows Phone 7 -käyttöjärjestelmää ja myöhemmin myös sen uudempia

versioita. Pääasialliseksi syyksi kerrottiin Symbianin heikko käytettävyys verrattuna kilpailijoihin ja Meegon myöhästymisen. Osana strategiaa on Symbian-laitteiden myynti ja kehitystyö vielä ainakin 2011 vuoden aikana ja laitteiden tukeminen jatkossakin, mutta pitkän tähtäimen suunnitelmana on siitä kokonaan luopuminen. (Digitoday 2011.)

Muut

BlackBerry on kanadalaisen Research In Motionin (RIM) valmistama mobiililaitte, joka on suosittu etenkin yrityskäytössä hyvän tietoturvasa ansiosta. BlackBerryn pääasiallinen markkina-alue on Pohjois-Amerikka. RIM pyrkii laajenemaan myös muille alueille ja kehittää nykyisin puhelimia, joita ei ole suunniteltu pelkästään yrityskäyttöön. Windows Phone 7 (WP7) on uusi versio Microsoftin pyrkimyksestä päästä mobiilimarkkinoille. Tällä hetkellä se edustaa marginaalivähemmistöä, mutta pyrkii nousemaan kolmen suosituimman joukkoon Nokian yhteistyön avustuksella. On mahdotonta sanoa, miten tilanne tulee kehittymään, ja tulokset nähdään vasta muutaman vuoden kuluttua. Muita marginaalivähemmistöjä ovat Palmin kehittämä, mutta nykyisin HP:n omistama WebOS ja Samsungin kehittämä Bada. Lisäksi tulossa on Nokian ja Intelin yhteistyö Meego, jonka piti alun perin olla Nokian seuraava käyttöjärjestelmä älypuhelimissa, mutta sen tulevaisuus on tällä hetkellä kysymysmerkki.

3.2 Mobiilisovellus

Mobiilisovelluksella tarkoitetaan mobiililaitteessa käytettävää ohjelmaa tai peliä. Sovellus voi olla myös esiasennettuna laitteeseen. Tyypillisiä esiasennuksia ovat esimerkiksi sähköposti- ja kalenterisovellukset. Ohjelmat ja pelit pyrkivät hyödyntämään älypuhelimien kattavia ominaisuuksia, kuten GPS-paikannusta, 3G-yhteyttä ja monikosketusta tukevaa näyttöä. Uudet ja kiinnostavat innovaatiot näyttelevät isoa osaa sovellusten suosiossa. (Webopedia n.d.)

3.3 Sovellusten jakelukanavat

Kolmannen osapuolen sovelluskauppoja on ollut olemassa tavalla tai toisella jo vuosikymmenen ajan. Aikanaan myynti perustui lähinnä soittoääniin, taustakuviin ja Java-sovelluksiin, joita toki myydään edelleenkin. Mobiilikäyttöjärjestelmien avautuessa kehittäjille mukaan tulivat myös natiivit ohjelmat. Kauppojen suosiota rajoitti kuitenkin vielä kehittymätön verkkotekniikka, kova kilpailu ja puute suoraan puhelimesta ostamisesta. Ohjelmat ladattiin hitaan verkon yli ja teleoperaattorit keräsivät suurimman osan voitoista. Lisäksi verkon käyttö oli hintavaa. Kolmannen osapuolen palvelut ovat edelleen voimissaan ja osa niistä on keskittynyt yhteen alustaan ja osa tarjoaa sisältöä useille eri järjestelmille. Niiden käyttäjäkunnat ovat kuitenkin usein murto-osassa verrattuna alustojen omiin natiiveihin palveluihin. Natiivienkin kauppojen välillä kaksinousevat selvästi muiden yläpuolelle suosiossa.

App Store

Mobiilisovellusten suosion kasvaminen ja muuttuminen ilmiöksi alkoi kesällä 2008, kun Apple julkaisi uuden iPhone 3G:n yhteydessä oman sovelluskaupansa, App Storen. Kun aikaisemmin muiden käyttöjärjestelmien ohjelmat ja pelit olivat olleet vapaana pitkin Internetiä, Applen strategia oli tuoda iOS:iin saatavat sovellukset yhteen paikkaan. App Storen julkaisun yhteydessä tarjolla oli noin 500 sovellusta. Osa oli ilmaisia ja osa maksullisia. (Markoff 2008.) Ensimmäisen viikon aikana ladattiin yli 10 miljoonaa sovellusta. Puoli vuotta myöhemmin latauksia oli kertynyt jo puoli miljardia ja vuoden kuluttua latauskerrat ylittivät 1,5 miljardia. Tätä kirjoitettaessa App Store on käytössä yli 90 maassa, tarjolla on yli 330 000 sovellusta ja latauksia on kertynyt yli kymmenen miljardia. (Slivka 2011.)

Android Market

Android Market lanseerattiin pian käyttöjärjestelmän julkaisun jälkeen loka-kuussa 2008. Kaupan selaamiseen luotu sovellus ("Market") on esiasennettuna useimmissa Android-laitteissa. Sen avulla käyttäjät pystyvät lataamaan ja ostamaan sovelluksia. Helmikuussa 2011 Google julkaisi Web-pohjaisen ohjelman, jolla käyttäjät pystyvät selaamaan ja asentamaan sovelluksia laitteisiinsa suoraan tietokoneelta. Jakelu-, osto-, ja myyntimahdollisuudet vaihtelevat eri maiden välillä. Android Market on käytössä 44 maassa, joista 32:ssa on mahdollista ostaa sovelluksia. Lopuissa on mahdollista ladata vain ilmaisia. Market hyväksyy ilmaiseksi jaettavia ohjelmia kehittäjiltä 147 maasta, mutta vain 29 maan asukkaat voivat luoda maksullisia ohjelmia jakeluun. Markkina-alueita ollaan kuitenkin koko ajan kasvattamassa. Marketin käyttö edellyttää Google-tiliä ja ohjelmien ostaminen tapahtuu Google Checkoutilla. (Android Market Help n.d.) Tätä kirjoitettaessa Android Market tarjoaa lähes 300 000 sovellusta ja latauskerrat ovat ylittäneet kolme ja puoli miljardia. (Androlib 2011.)

Muut

Muut jakelukanavat edustavat vielä pientä osaa, jos tarkastellaan ohjelmien ja latauksien määrää. Kaksi seuraavaksi suurinta kauppaa ovat Nokian Ovi Store ja Black Berryn AppWorld. Ensiksi mainittu sisältää noin 44 000 sovellusta ja latauksia kertyy noin neljä miljoonaa päivässä. (Forum Nokia 2011.) Jälkimmäisenä mainitun luvut ovat 16 000 ja 1,5 miljoonaa päivässä. (BlackBerry 2011.) Ainakin toistaiseksi marginaalivähemmistöä edustavat Microsoftin Windows Phone Marketplace ja WebOS:n App Catalog.

4 MOBIILIALUSTAN VALITSEMINEN

Kun idea oman mobiilisovelluksen kehittämiseksi nousee esille, ensimmäisenä kysymyksenä on epäilemättä ”Mistä aloittaa?”. Kehittäminen alkaa mobiilialustan valinnalla. Valittavana on useita järjestelmiä, eikä väärää valintaa ole olemassa, mutta helpointa lienee aloittaa jollakin suosituimmista alustoista. Toki kukaan ei estä kehittämästä useille alustoille yhtä aikaa, mutta helpompaa on keskittyä yhteen kerrallaan. Valitsemisperusteita voi olla useita, kuten järjestelmän tai sen sovelluskaupan suosio, ohjelmointikieli, käyttöönoton helppous ja kehittämisen kustannukset. Moni itsenäinen kehittäjä valitsee alustan oman puhelimensa perusteella, koska haluaa nähdä luomuksensa juuri omassa käytössä.

Tässä luvussa tarkastellaan lyhyesti valittujen mobiilikäyttöjärjestelmien ohjelmointityökaluja, vertaillaan niiden käyttöönottoa ja käydään läpi yleisellä tasolla vaatimuksia sovelluskehityksen aloittamiseen. Kaikilla alustoilla on olemassa sovelluskehittäjille omistettu osio WWW-sivuillaan, joka sisältää ohjeita alkuun pääsemiseen. Näitä ohjeita hyödyntämällä kokeilen asentaa omalle koneelleni tarvittavat työkalut ja luomalla yksinkertaisen Hello World -sovelluksen. Kyseinen ohjelma ei varsinaisesti sisällä mitään ominaisuuksia, joten sen perusteella ei voida luoda johtopäätöksiä ohjelmointikielen vaikeustasosta. Sen sijaan sen avulla voidaan testata työkaluja ja alkuunpääsyn helppoutta.

Mobiilisovellusten, kuten perinteistenkin sovellusten, ohjelmointi alkaa tarvittavien työkalujen lataamisella. Minimivaatimuksena ovat yleensä alustan kehittäjien luoma SDK (Software Development Kit), joka on kokoelma tarvittavia työkaluja ja IDE (Integrated Development Environment) eli ohjelmointiympäristö. Lisäksi vaatimuksena saattaa olla ylimääräisiä työkaluja, kuten Androidin tapauksessa JDK (Java Development Kit).

4.1 Symbian, QT ja niiden tulevaisuus

Koska Symbian on ollut pisimpään markkinoilla, on sille kertynyt eniten eri versioita, käyttöliittymiä ja puhelinmalleja. Se ei alun perin sisältänyt käyttöliittymää ollenkaan, joten eri valmistajat kehittivät niitä itse. Tämän vuoksi Symbianin versiot tunnetaan paremmin käyttöliittymän nimellä ja versiolla. Tunnetuin ja laajimmalle levinnyt näistä on Nokian S60. Tätä edelsivät Series 80 ja Series 90, jotka olivat käytössä lähinnä vanhoissa Communicator-malleissa. Tällä hetkellä versioita, joille sovelluksia kehitetään, on kolme: S60 3rd Edition, jota hyödyntävät ei-kosketusnäytölliset puhelimet, edistyneempi S60 5th Edition (tunnetaan myös nimellä Symbian^1) ja uusin Symbian^3.

3rd ja 5th Edition muistuttavat käyttöliittymiltään ja ominaisuuksiltaan toisia ja niitä käytetään vielä aktiivisesti uusissakin puhelinmalleissa, mutta teknologia alkaa olla vanhentunutta. Sen sijaan Symbian^3 on kokonaan uusi tuote ja kohdistettu tulevaisuuden puhelimille. Käyttöliittymä on suunniteltu uudeksi ja mukaan on tuotu uusia ja kriittisen tärkeitä ominaisuuksia, kuten tuki kapasitiivisten kosketusnäyttöjen monikosketukselle. Kehittäjän näkökulmasta eri versiot saattavat rajoittaa sovelluksen ulottuvuutta. Toiminnallisesti yksinkertaiset ohjelmat voidaan toteuttaa kaikille versioille, jolloin potentiaalisia käyttäjiä on monikertainen määrä, mutta esimerkiksi monikosketusta hyödyntävä peli on mahdollista vain Symbian^3:ssa. Lisäksi 5th ja 3rd Editionia hyödyntävät puhelimet ovat tehoiltaan melko vaatimattomia älypuhelimien standardeilla. Symbianilla on laajin kirjo eri laitteita, mikä aiheuttaa Androidin tavoin ongelmia sovellusten toimivuudessa laitteiden välillä.

Symbian on kirjoitettu C++ -ohjelmointikielellä ja nykyään myös alustan sovellukset kirjoitetaan pääasiassa samaisella kielellä. Ennen Qt:n tuloa sovellukset kirjoitettiin muokatulla C++ -kielellä. Se poikkesi alkuperäisestä mobiililaitteille kustomoiduilla kirjastoilla. Näiden opettelua ja ohjelmointia pidettiin yleisesti hankalana ja laitteiden kehittyessä tämä nähtiin myös haittaavana tekijänä, koska kieli oli alun perin kehitetty aikana, jolloin puhelimet olivat rajoittuneempia. Symbianille voi myös edelleen kehittää tällä menetelmällä, mutta vuoden 2010 alusta lähtien kehitys on virallisesti tapahtunut Qt:n avulla. Symbian tukee myös muita kieliä, kuten Java ME, Flash Lite ja Python. (Qt 2011.)

Qt (lausutaan ”cute”) on alustariippumaton ohjelmistojen ja graafisten käyttöliittymien kehitysympäristö, joka perustuu avoimeen lähdekoodiin. Käytännössä Qt:n käyttö mahdollistaa yhden koodin hyödyntämisen monella eri alustalla. Menetelmä on käytännöllistä ja aikaa säästävää. Jos kehittäjä haluaa esimerkiksi luoda sovelluksen, joka toimii S60 3rd ja 5th Editionissa, Symbian^3:ssa sekä Meegossa, ei sitä tarvitse koodata neljästi neljän eri SDK:n avulla. Lisäksi samainen sovellus toimii haluttaessa myös vaikkapa Windows- ja Mac-tietokoneissa. Qt:ssa on sisäänrakennettu tuki C++ -kielelle, joka on ideaali natiivien C-pohjaisten ohjelmien luomiseen, mutta sitä voidaan käyttää myös muilla kielillä eri kirjastojen avulla. Qt:n avulla on kehitetty mm. Google Earth ja Skype. Kehityksestä vastaa nykyisin Nokian omistama Qt Development Frameworks. (Qt 2011.)

Ennen kuin yhteistyökuviot Microsoftin kanssa lanseerattiin, Nokian älypuhelinstrategiana oli Symbianin ja Meegon hyödyntäminen. Sovelluskehittäjät opastettiin Qt:n käyttöön ja vakuuteltiin sen olevan tulevaisuus ja standardi. Nyt tilanne on kuitenkin muuttunut. Symbian tullaan ajamaan alas muutaman vuoden sisällä ja Meego jäänee marginaalikokeiluksi. Lisäksi WP7 -kehitys tapahtuu Microsoftin omilla työkaluilla, ei Qt:lla. Myös Nokian sovelluskauppa Ovi Store tullaan sulauttamaan Microsoftin omaan Phone Marketplaceen. On totta, että Symbianilla varustettuja puhelimia on tällä hetkellä enemmän kuin yhdelläkään kilpailijalla, ja tämä tilanne tulee vielä ainakin hetken aikaa säilymään. Toisaalta faktaa on myös se, että WP7 on täysin marginaaliasemassa. Itse näen tilanteen olevan tällä hetkellä niin auki, etten välttämättä hyppäisi kehittämään kummallekaan ainakaan ensimmäisenä, vaikka markkinoita olisi-kin Symbianin kohdalla ja tulevaisuus teoriassa näyttäisikin valoisalta WP7:n kanssa.

4.2 Androidille kehittäminen pähkinäkuoressa

Androidin sovellukset kirjoitetaan oliopohjaisella Java-ohjelmointikielellä, mutta niitä ei kuitenkaan ajeta perinteisessä Java-virtuaalikoneessa, vaan Androidin omalla Dalvik-virtuaalikoneella. Dalvik on suunniteltu erityisesti mobiili-

laitteita varten. Ohjelmat, jotka on tehty toimimaan Javan virtuaalikoneissa, eivät toimi natiivina Androidissa. Google myös tarjoaa mahdollisuuden luoda natiiveja ohjelmia NDK:n (Native Development Kit) avulla. Tällöin ohjelmointi tapahtuu C/C++ -kielellä ja sovellukset toimivat suoraan Linux-ytimessä virtuaalikoneen sijaan. Natiivi sovelluskehitys ei välttämättä tarjoa parempaa suorituskykyä ohjelmille, mutta sen avulla voidaan luoda kompleksikkaampia ohjelmia. Google kuitenkin opastaa yleisellä tasolla ainakin aloittelijoita käyttämään Javaa. Ulkoasun muodostamiseen käytetään XML:ää (Extensible Markup Language), jolloin aikaisempi kokemus Internet-sivujen luomisesta voi olla hyödyksi. (Android Developers 2011.)

Androidin kehitysympäristö on avoin, mikä tarkoittaa sitä, että kehittäjillä on täysi pääsy samoihin rajapintoihin joita järjestelmäsovellukset eli nk. ydinohjelmat käyttävät. Ydinohjelmaksi lasketaan esimerkiksi puhelutoiminnot, tekstiviestit ja osoitekirja sekä muut vakiona mukana tulevat sovellukset. Kaikille ohjelmille yhteiset rajapinnat tarkoittaa, että kolmannen osapuolen kehittäjät voivat luoda rikkaita ohjelmia pääsemällä käsiksi mm. puhelimen laitteistoon, sijaintitietoihin, datayhteyksiin, hälytyksiin, tekstiviestiominaisuuksiin ja ilmoitusvalikkoon (Notification bar). Androidin arkkitehtuuri kannustaa jakamaan resursseja, prosesseja ja dataa muiden ohjelmien kesken. Varsinaisesti avoin ympäristö ei tuo mitään erityisen ylivoimaista etua verrattuna esimerkiksi Applen suljettuun ympäristöön, koska myös iOS:ille voi luoda yhtä rikkaita sovelluksia. (Android Developers 2011.)

4.2.1 Koodiesimerkki

Android-sovelluksen kehittäminen alkaa Android SDK:n lataamisella. Se tukee yleisimpiä käyttöjärjestelmiä, kuten Windows XP ja sitä uudemmat, Mac OS X ja Linux. Kuten käytännössä kaikki SDK:t, myös Androidin on ilmainen ladata ja käyttää. Kehittämisestä ei myöskään peritä maksuja. Ohjelmointiympäristönä käytetään Androidin suosittelemaa avoimen lähdekoodin Eclipseä. Lisäksi pakollisena on JDK 5 tai JDK 6 (Java Development Kit), joka täytyy olla asennettuna, ennen kuin SDK suostuu toimimaan. Suositeltavaa on myös asentaa ADT (Android Development Tools) -liitäntäinen Eclipseen, joka tarjoaa kehit-

tämistä helpottavia lisätyökaluja. Asennusjärjestyksenä on ensimmäisenä Eclipse, JDK, SDK ja lopuksi ADT. (Android Developers 2011.)

Kuviossa 2 on yksinkertainen ”Hello Android” -sovellus, joka perustuu Activity-luokkaan. Se on yksittäinen kokonaisuus, jonka avulla voidaan suorittaa toimintoja ja se on olennainen osa arkkitehtuuria. Androidin käyttöliittymät muodostuvat View-olioista. Niitä ovat esimerkiksi napit, kuvat ja tässä tapauksessa (TextView) tekstikehys. setText() määrittelee tekstikehysen ja lopuksi setContentView kutsuu sen käyttöliittymään. Sovellusta voidaan testata Android Virtual Device -simulaattorissa. (Android Developers 2011.)

```
package com.android.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}
```

KUVIO 2. Hello World -sovellus Androidissa

Kaiken kaikkiaan työkalujen lataamisesta ohjelman näkymiseen simulaattorissa kului aikaa vähän reilu tunti. Sivuston ohjeet ovat selkeät ja sisältävät paljon hyödyllisiä esimerkkejä. Koska taustallani on hieman Java-ohjelmointia, tuntui koodin kirjoittaminen ja logiikka alusta alkaen selvähköltä. Seuraavat esimerkit sivustolla käsittelevät XML:n käyttöönottoa ulkoasun muokkaamiseen HelloAndroid -sovelluksessa, joka mielestäni sopii hyvin jatkoksi.

4.2.2 App Inventor

App Inventor on Googlen kehittämä ilmainen selaimessa toimiva ohjelma, jonka avulla kuka tahansa voi luoda sovelluksia kirjoittamatta yhtäkään riviä koo-

dia. Kyse on visuaalisesta ohjelmoinnista, jossa käyttäjät rakentavat sovelluksen erilaisista objekteista. Prosessi muistuttaa palapelin kokoamista. App Inventor on helppo omaksua ja siinä pääsee melko nopeasti alkuun. Vaikka ominaisuudet ovatkin kattavat ja käyttäjillä on pääsy esimerkiksi laitteen GPS:iin, ei sen avulla voida loppujen lopuksi luoda erityisen monimutkaisia sovelluksia. Sen sijaan se on oiva työkalu ohjelmoinnin vasta-alkajille. Se auttaa ymmärtämään ohjelmointia ja Android-järjestelmää. Vahvistamattomien tietojen mukaan myös Apple olisi tuomassa markkinoille vastaavan ohjelman iOS:ille. (App Inventor 2011.)

4.2.3 Haasteet

Koska Android on suhteellisen tuore tapaus, se kasvaa ja kehittyy isoin harppauksin. Uudet versiot tuovat mukanaan täysin uusia ominaisuuksia ja mahdollisuuksia kehittäjille. Päivitystahti tällä hetkellä on melko tiheä. Jokainen Android-laite on teoriassa mahdollista päivittää uusimpaan versioon, mutta päivityksen saatavuudesta vastaa aina laitevalmistaja ja/tai teleoperaattori. Useat laitteet sisältävät esimerkiksi kustomoituja käyttöliittymiä, jolloin päivitystä ei voi asentaa, ennen kuin laitevalmistaja on varmistanut yhteensopivuuden. Päivityksen saanti on kuluttajan näkökulmasta usein arvailujen varassa. Suosituimmat puhelinmallit ovat yleensä hyvin tuettuja, mutta vanhempien mallien omistajat eivät välttämättä ole yhtä onnekkaita. Lisäksi laitevalmistajat tuovat päivityksenä tarjolle usein vasta kuukausien päästä uuden Android-version julkaisusta. On siis olennaista miettiä versioyhteensopivuutta sovellusta luotaessa. Ongelmallista on myös laitteiden laaja kirjo. Markkinoilla on runsas määrä eri prosessoreilla, grafiikkapiireillä, muistimäärillä ja resoluutioilla varustettuja malleja. Tämän vuoksi sovellus ei välttämättä toimi yhtä hyvin/sulavasti kaikilla laitteilla, jolloin kehittäjä voi joutua luomaan mallikohtaisia parannuksia koodiin yhteensopivuuden takaamiseksi. (Android Developers 2011.)

Android on rakennettu toimimaan siten, että sovellukset on mahdollista luoda toimimaan kehitysversiota vanhemmissa ja uudemmissa. Android-version päivityessä ohjelmointirajapintaan (API) yleensä vain lisätään ominaisuuksia

minkä vuoksi eteenpäin yhteensopivuus on taattua. Päinvastoin vanhemmat versiot eivät tue uudempia ominaisuuksia. Näiden ominaisuuksien selvittäminen on kehittäjän vastuulla. Tämä onnistuu API Levelin avulla. Jokaisella Android versiolla on oma API Level. Androidin Developer -sivustolta voidaan tarkistaa, mikä ominaisuus toimii missäkin. Jos esimerkiksi luokan kohdalla lukee ”Since Api Level 3”, voidaan sitä käyttää Android 1.5 -versiossa ja luonnollisesti sitä uudemmissa. Toki on myös mahdollista unohtaa vanhemmat versiot täysin ja keskittyä vain tämän hetkiseen. (Android Developers 2011.)

TAULUKKO 1. Android -versioiden jakautuminen.

Versio	API Level	Jakautuminen
Android 1.5	3	3,9 %
Android 1.6	4	6,3 %
Android 2.1	7	31,4 %
Android 2.2	8	57,6 %
Android 2.3	9	0,8 %

Taulukossa 1 on listattu Androidin versiot ja niitä vastaavat API Levelit sekä eri versioiden jakautuminen markkinoilla. Tieto jakautumisesta on kerätty seuraamalla kaikkia Android Marketissa vierailevia laitteita aikavälillä 19.1.2011 - 9.2.2011. Tästä voidaan päätellä, että vanhempia kuin 1.5 versiolla varustettuja laitteita on käytännössä olematon määrä markkinoilla ja niitä ei kannata ottaa huomioon sovellusta luotaessa. Android päivittyi versiosta 1.6 suoraan versioon 2.0 (API Level 5), 2.0.1 (API Level 6) tuli lähes heti perässä ja kolmen kuukauden päästä ilmestyi 2.1. Koska nämä versiot ehtivät olla niin vähän aikaa ulkona, laitevalmistajat eivät ehtineet hyödyntää niitä laitteisiinsa, minkä vuoksi niitä ei myöskään näy taulukossa. (Android Developers 2011.)

4.3 iOS:ille kehittäminen pähkinänkuoressa

Toisin kuin Android ja Symbian, iOS ei ole avointa lähdekoodia. Sen ydin perustuu avoimeen Darwin BSD-käyttöjärjestelmään ja mukana on tiettyjä Mac OS X:n komponentteja, mutta itse järjestelmä on suljettu. Sovellukset tehdään Objective-C -kielellä. iOS SDK sisältää Xcode-työkalupaketin ja iOS-

simulaattorin sovellusten testaamiseen. Xcode sisältää kaikki tarvittavat työkalut sovellusten kehittämiseen, kuten ohjelmointiympäristön ja ohjelman graafisen käyttöliittymän luomiseen. SDK on mahdollista ladata ilmaiseksi rekisteröitymällä Apple kehittäjäksi. Sovelluksen voi kehittää ilmaiseksi, mutta jos sitä haluaa kokeilla laitteessa (ja myöhemmin julkaista App Storessa), on kehittäjän pakko liittyä maksulliseen iOS Developer Programiin. Yksityiskäyttäjälle se kustantaa 99 Yhdysvaltain dollaria vuodessa. Lisäksi rajoittavana tekijänä on se, että iOS SDK on saatavilla ainoastaan Mac-tietokoneille, joissa käyttöjärjestelmänä on Mac OS X Leopard tai sitä uudempi versio. (iOS Dev Center 2011.)

4.3.1 Koodiesimerkki

Kun Xcodessa tehdään uusi projekti, ohjelma luo valmiiksi tiedostoja, jotka sisältävät koodia. Varsinaiset muutokset nähdään kuvioissa 3 ja 4. Vaikka koodi onkin erilaista Androidin esimerkkiin verrattuna, toimintaperiaate on käytännössä sama. MyView on luokka, joka piirtää halutun tekstin näytölle ja se kutsutaan käyttöliittymään HelloWorldAppDelegate.m:ssä. (iOSDev Center 2011.)

Kokeilut suoritettiin toimeksiantajalta lainatulla MacBookPro:lla. Sovellustyökalujen käyttöönotto on vähintään yhtä helppoa kuin Androidissa, ja Developer-sivusto on myös erittäin kattava ja sisältää paljon havainnollistavia esimerkkejä. Ohjelmointikieli tuntui hieman mystiseltä vähäisen ohjelmointikokemuksen vuoksi, ja tässäkin raapaistiin vain pintaa. Sen sijaan Xcoden käyttö tuntui alusta lähtien helpolta. Oppimiskäyrä on epäilemättä Objective-C:n kohdalla pisimpiä.

```
#import "HelloWorldAppDelegate.h"
#import "MyView.h"

@implementation HelloWorldAppDelegate

@synthesize window;

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
```

```

    // Override point for customization after application launch
    MyView *view = [[MyView alloc] initWithFrame:[window frame]];
    [window addSubview:view];
    [view release];

    [window makeKeyAndVisible];
    return YES;
}

- (void)dealloc {
    [window release];
    [super dealloc];
}

@end

```

KUVIO 3. HelloWorldAppDelegate.m -tiedosto.

```

- (void)drawRect:(CGRect) rect {
    NSString *hello = @"Hello, World!";
    CGPoint location = CGPointMake(10, 20);
    UIFont *font = [UIFont systemFontOfSize:24.0];
    [[UIColor whiteColor] set];
    [hello drawAtPoint:locationwithFont:font];
}

```

KUVIO 4. MyView.h -tiedosto.

4.3.2 iOS-laitteet

Kuten muutkin mobiilikäyttöjärjestelmät, myös iOS kehittyy jatkuvasti ja päivitystahti on ripeä. Tällä hetkellä käytössä ovat laajalti versiot 3.x ja 4.x. App Store ei tue iOS 3.x:ää vanhempia versioita, joten niitä ei tarvitse ottaa huomioon. Apple ei julkaise Androidin tapaan tilastoja versioiden suosiosta, minkä vuoksi osuudet ovat enemmän tai vähemmän arvailujen varassa. Sen sijaan kehittäjiä on mahdollista saada tilastoja heidän ohjelmiansa lataajista. Kaksi isohkoa tekijää ilmoittaa, että heidän sovellustensa lataajista noin 90 % käyttää versiota 4.x. (Perez 2011.) Tämä luku ei missään nimessä tarjoa koko totuutta, mutta sitä voidaan pitää suuntaa antavana. Pitää muistaa, että osa käyttäjistä ei osaa päivittää tai jaksa vaivautua päivittämään laitteitansa. Nelosversio sisältää kriittisen ominaisuuden vanhempiin verrattuna: se tukee ohjelmien moniajtoa.

iOS:ille kehittämistä helpottaa vähäinen laitteiden määrä verrattuna Androidiin ja Symbianiin, mikä taas käytännössä säästää aikaa ohjelmoinnissa ja testauksessa. iPhoneen lisäksi käyttöjärjestelmää hyödyntävät mediatoistin iPod Touch ja taulutietokone iPad. Lisäksi kaikki em. laitteet on mahdollista päivittää vähintään 3.x versioon, mikä takaa kehittäjälle erittäin laajan laitekannan (taulukko 2), mikäli sovellus pystytään toteuttamaan kaikkien laitteiden ominaisuuksien rajoissa. Luonnollisesti ohjelma ei voi perustua ominaisuuksiin, mitä vanhimmissa laitteissa ei ole. Esimerkiksi ensimmäinen iPod Touch ei sisällä kameraa ja alkuperäinen iPhone ei tue 3G-yhteyksiä. (Van Zummeren 2010.)

TAULUKKO 2. Käyttöjärjestelmäversioiden saatavuus iOS -laitteissa.

Malli	Uusin tuettu iOS-versio
iPhone	3.1.3
iPhone 3G	4.2.1
iPhone 3G S	4.3 Beta 2
iPhone 4	
iPodTouch 1st generation	3.1.3
iPodTouch 2nd generation	4.2.1
iPodTouch 3rd generation	4.3 Beta 2
iPodTouch 4th generation	
iPad WiFi & iPad WiFi + 3G	4.3 Beta 2

iPod Touch on käytännössä sama laite kuin iPhone, mutta siitä on riisuttu matkapuhelinominaisuudet. Tämän vuoksi iPhonele tehdyt sovellukset toimivat Touchissa ilman lisätoimenpiteitä. Uudet mallit ovat kehittyneet myös laite-tehoiltaan ja ominaisuuksiltaan. Kaksi ensimmäistä sukupolvea iPhoneista ja iPod Touchista alkavat olla tekniikan puolesta jo vanhentuneita, mikä rajoittaa esimerkiksi pelien grafiikoita ja sulavuutta. Uusimmassa iPhoneissa ja iPod Touchissa on neljä kertaa edeltäjänsä tarkempi Retina-näyttö. Isompi resoluutio ei aiheuta koodiin muutoksia, sillä iOS osaa skaalata käyttöliittymän perustyökälu (napit, valikot, sliderit yms.) näytölle sopivaksi. Sen sijaan grafiikka, joka on alun perin suunniteltu vanhemmille näytöille, saattaa näyttää rumalta. (Van Zummeren 2010.)

Taulutietokone iPad eroaa joukosta isolla näytöllään ja mahdollisuudella luoda monimutkaisempia käyttöliittymiä. Tämä johtuu siitä, että näytöllä on mahdollista olla esillä useampia näkymiä, kun iPhone/iPod Touch joutuvat tyytymään yhteen näkymään. Useimpien iPhonelle tehtyjen ohjelmien pitäisi toimia myös suoraan iPadissa. Kaikki ohjelmat eivät vain välttämättä skaalaudu koko näytön kokoiseksi. Sovellukseen on mahdollista sisällyttää eri käyttöliittymä iPadia varten kierrättämällä koodia. App Storessa tällaisten ohjelmien perässä on plus-merkki, jolloin käyttäjät tietävät, että se toimii molemmissa laitteissa. Tällä on kaksi etua: käyttäjien tarvitsee ostaa ohjelma vain kerran mikä lisää asiakastytyväisyyttä ja kehittäjän tarvitsee ylläpitää vain yhtä versiota ohjelmasta. Toki markkinoilla on useita sovelluksia, joista on eri versiot eri laitteille tai vain tietyille laitteelle. (Van Zummeren 2010.)

4.4 Selainpohjaiset sovellukset

Tekniikan kehittyessä selainpohjaisten sovellusten merkitys kasvaa. Ne ovat alustariippumattomia ja toimivat jokaisessa laitteessa, jossa on yleisimpiä standardeja (HTML, CSS, JavaScript) tukeva selain. Tällä hetkellä sellainen on käytössä tai saatavilla lähes jokaiseen mobiilikäyttöjärjestelmään. Selainpohjaisten sovellusten teko on jo nyt mahdollista ja esimerkiksi webOS:ssa kaikki kehitys on selainpohjaista. Lisäksi BlackBerry opastaa kehittäjiään siihen. Vielä nykyisillä standardeilla ja tekniikalla ei välttämättä ole mahdollista saada aikaan kaikista näyttävimpiä sovelluksia, mutta tuleva HTML5 tuo mukanaan uudet rajapinnat, jotka avaavat kokonaan uusia ja rikkaita mahdollisuuksia. Lisäksi Web-pohjaisten sovellusten kehitystä pidetään helpommin lähestyttävänä kuin perinteisten asennettavien ohjelmien kehitystä. Selainpohjaisten sovellusten tärkeyttä ja suosion suuruutta on kuitenkin vielä tässä vaiheessa hankala arvioida. (W3C, 2011.)

4.5 Yhteenveto

Kuten aikaisemmin todettiin, alustan valinnassa ei ole olemassa väärää valintaa. Ohjelmoinnin näkökulmasta kokenut olio-ohjelmoija oppii epäilemättä nopeasti kielen kuin kielen. Kun verrataan Androidia ja iOS:a, on ensinnä mainitun puolella moni asia. Työkalut ovat paremmin saatavilla ja julkaisun kustannukset ovat huomattavasti pienemmät. Jos kehittäjä on kuten kirjoittaja eli aloittelija, jolla on vain vähän ohjelmointikokemusta, Java lienee helpompi omaksua kuin Objective-C. Javalla voi luoda ohjelmia Androidin lisäksi myös BlackBerryyn ja sen osaaminen helpottaa JavaScriptin opettelua, mikä saattaa olla hyödyksi selainpohjaisten sovellusten yleistyessä. Vastaavasti iOS:a puoltaa pieni laitevalikoima, joka helpottaa ohjelmointia ja lisäksi Applen sovelluskauppa on ylivoimainen markkinajohtaja. Molempien alustojen ohjelmointiympäristöjen ja työkalujen käyttöönotto on suurin piirtein yhtä helppoa. Opetusmateriaalista ei ole pulaa, sillä oppaita eri järjestelmille ovat kaupat ja Internet pullollaan.

TAULUKKO 3. Yhteenveto mobiilialustoista.

	Ohjelmointikieli	Työkalut, ympäristöt	Kehittämisen maksullisuus	Lopullinen hinta	Käyttöjärjestelmät
Android	Java, C/C++	Eclipse JDK	Ei	25USD	Windows Mac OS X Linux
iOS	Objective-C	Xcode	Ei	99 USD /vuosi	Mac OS X
Symbian	Qt (C/C++)	Nokia Qt	Ei	1 €	Windows Mac OS X Linux
WP7	.NET	XNA Silverlight Visual Studio Visual Basic	Ei	99 USD /vuosi	Windows
BlackBerry	Web-standardit Java	WebWorks, Eclipse,	Ei	Ilmainen toistaiseksi	Windows Mac OS X Linux
webOS	Web-standardit	SDK Eclipse	Ei	Ilmainen toistaiseksi	Windows Mac OS X Linux
MeeGo	Qt (C/C++)	Nokia Qt	Ei	1 €	Windows Mac OS X Linux

5 SOVELLUKSEN JULKAISU

5.1 Allekirjoitus

Ennen sovelluksen julkaisua se tulee allekirjoittaa digitaalisesti. Allekirjoitus hyödyntää salaustekniikkaa ja sen avulla pyritään varmistamaan koodin aitous, eheys ja alkuperä. Varmistus ei kuitenkaan tarkoita automaattisesti, että koodi olisi *luotettava*. Allekirjoitus ei ole sama asia kuin kopiosuojaus. Käytännössä kaikki mobiililaitteet vaativat toimivan allekirjoituksen ennen kuin sallivat sovelluksen asentamisen, mutta joidenkin laitteiden kohdalla se on mahdollista ottaa pois käytöstä. Sovelluskaupat eivät kuitenkaan hyväksy allekirjoittamattomia ohjelmia. Viestintävirasto määrittelee asian seuraavasti:

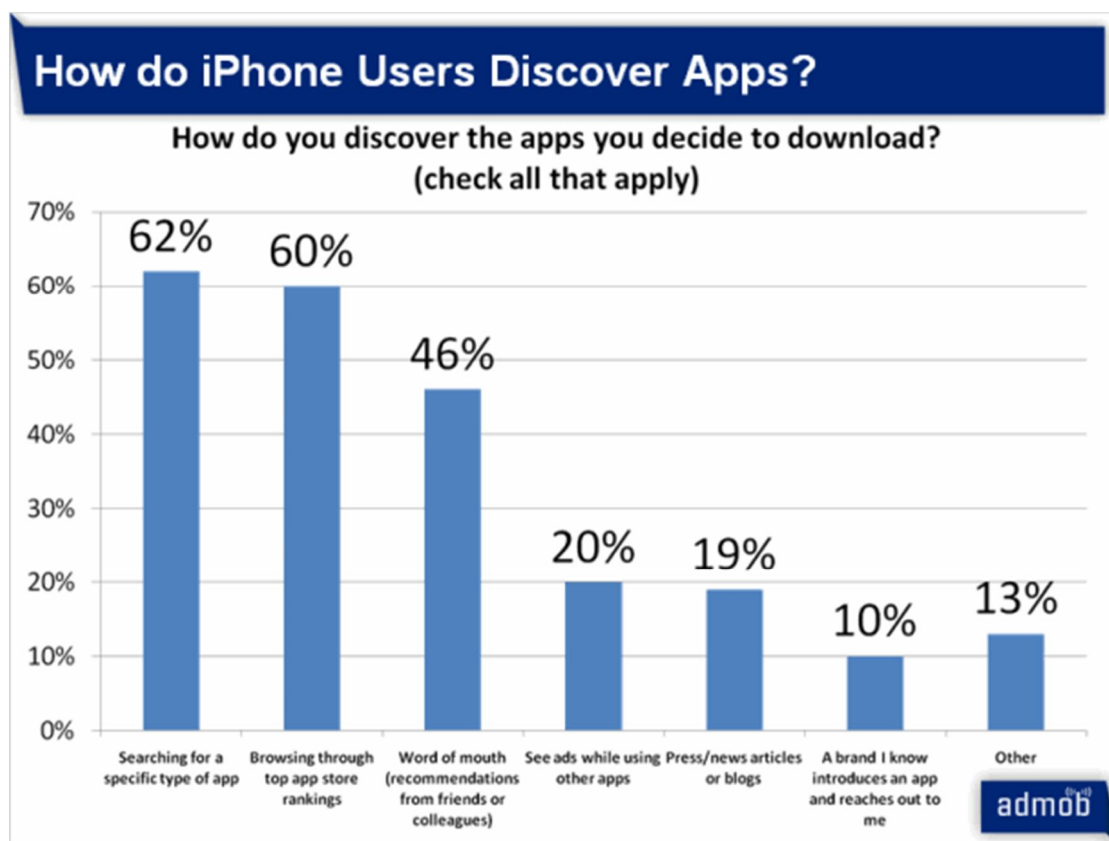
Digitaalisen allekirjoituksen tehdessään allekirjoittaja laskee allekirjoitettavasta viestistä yksisuuntaisen funktion avulla tiivisteen, jonka hän salaa yksityisellä avaimellaan. Vastaanottaja laskee saamastaan viestistä myös tiivisteen ja vertaa sitä allekirjoittajan varmenteesta saadulla julkisella avaimella avaamaansa salattuun tiivisteeseen. Mikäli tiivisteet ovat samat, on viesti kulkenut ehyenä (muuttumattomana) ja sen on allekirjoittanut juuri se henkilö, joka varmenteessa mainitaan. (Viestintävirasto, 2007.)

Androidin kohdalla sovelluksen allekirjoituksen voi hoitaa itse ilmaiseksi saatavilla työkaluilla. Dokumentointi löytyy Android Developer -sivustolta. Symbianin ja iOS:n kohdalla allekirjoitus ei onnistu itse, sillä niiden kohdalla sertifikaatti pitää tulla toiselta osapuolelta. Tähän on olemassa lukuisia palveluita, mutta helpointa lienee käyttää hyväksi alustan itsensä tarjoamia ratkaisuja. Nokia ja Apple tarjoavat sitä ilmaiseksi, mutta edellyttävät ensin kehittäjäksi rekisteröitymistä.

5.2 Ohjelmakuvaus

Julkaisua varten on tarjottava erilaista dataa kuvailemaan sovellusta. Kuvailut vaihtelevat kauppojen välillä, mutta tyypillisesti mukaan tulee laittaa kuvan-kaappauksia ja kuvake ohjelmavalikkoa varten. Lisäksi mukaan tulee kirjoittaa

lyhyt esittely ohjelmasta, erilaista metadataa ja yhteystiedot tukea/palautetta varten. Ohjelmakuvaus on osana markkinointia ja toimii myös mainospuheena. Erityisesti avainsanoja kannattaa miettiä, sillä Admobin kyselyn mukaan suurin osa iPhoneen käyttäjistä löytävät uusia sovelluksia hakemalla. Avainsanoja mietittäessä on hyvä tutustua valitun sovelluskaupan hakuun ja etsiä esimerkiksi kilpailevia sovelluksia. Hakusanat eivät ole yleensä näkyvillä eikä niitä voida suoraan "lainata" kilpailijalta, mutta tutkimustyö helpottaa ymmärtämään haun toimintaa. Applen kohdalla avainsanoja moderoidaan tarkastusprosessissa ja niiden epäoleellinen käyttö voi johtaa sovelluksen hylkäykseen. Myös Android Market moderoi niitä käyttäjien ilmoitusten perusteella. Yleisten sanojen liikkakäyttöä kannattaa välttää, vaikka ne liittyisivätkin ohjelmaan. Yleiset sanat (kuten "App", "Game", "Fun") tuottavat hakutuloksissa pitkän listan sovelluksia, johon on helppo hukkuu. Vastaavasti uniikit avainsanat tuottavat lyhyen listan, jolloin näkyvyys paranee. (Wooldridge & Schneider 2010. 309-310.)



KUVIO 5. Kuinka iPhoneen käyttäjät löytävät uusia sovelluksia? (Lardinois 2009.)

Kun käyttäjä saapuu sovelluksen sivulle, hänen osto/latauspäätökseensä vaikuttaa kolme tekijää: ohjelman esittely, kuvankaappaukset ja muiden käyttäjien arviot. Esittelyssä tulee huomioida mobiilimaailman kirjoittamattomat säännöt. Sisällön tulisi olla lyhyt ja ytimekäs, koska useimmat käyttäjät ovat tottuneet nopeasti silmäilemään näkemänsä. Pitkät kappaleet vaativat runsaasti näytön vierittämistä, mikä voi johtaa turhautumiseen ja lopulta sivulta pois siirtymiseen. Jos myyntipuheessa ei tule selvästi esille ohjelman tarkoitus, se saattaa helposti karkoittaa käyttäjän. Sanotaan, että kuva kertoo enemmän kuin tuhat sanaa ja se pitää paikkansa myös tässä kontekstissa. Kuvan tulisi vakuuttaa käyttäjä ja olla yhteistyössä esittelyn kanssa. Hyvä kuvankaappaus näyttää jonkin toiminnallisuuden/ominaisuuden sovelluksesta. Vaikka kuvankaappaukset ovatkin yleisin muoto, ei kuvaa ole rajoitettu tähän. Se voi myös olla tilannekuva, jossa sovellusta käytetään. (Wooldridge & Schneider 2010. 315.)

5.3 Tarkastusprosessi

Android Market ei sisällä sovelluksen tarkastusta ollenkaan, ja ohjelmat näkyvät siellä lähes välittömästi lähetyksen jälkeen. Kehittäjä joutuu kuitenkin hyväksymään ehdot sisällöstä. Tiivistettynä niissä sanotaan, että sovellus ei saa sisältää mitään laitonta. Marketin sisältöä valvotaan ilmiannoilla. Jos käyttäjä huomaa laitonta/kyseenalaista materiaalia tai huonosti toimivan ohjelman, voi hän ilmoittaa siitä ylläpidolle, joka tarvittaessa puuttuu asiaan ja poistaa ohjelman. Esitarkastuksen puuttuminen on aiheuttanut kehuja ja kritiikkiä. Toiset pitävät siitä, ettei mikään taho ole rajoittamassa tarjontaa. Toisten mielestä Android Market pursuaa huonosti tehtyjä sovelluksia ja on avoinna haittaohjelmille. Huomionarvoisena seikkana voidaan pitää myös sitä, että koska Android Market toimii Yhdysvalloissa, voidaan sieltä ladattuja ohjelmia pitää Yhdysvaltojen vientituotteina. Tällöin ne voidaan tarvittaessa altistaa paikallisille vientilaeille. Kyseinen lakiviidakko on valtava ja sen tulkitseminen on kokonaan oma aiheensa. Sisältöä ei kuitenkaan kannata stressata liikaa ja mallisjärki on hyvä ystävä tässäkin asiassa. (Android Market Help n.d.)

App Store sen sijaan toteuttaa laaduntarkastuksen jokaiselle julkaisulle. Tässä prosessissa kokeillaan sovelluksen päätoiminnot ja tarkastetaan sisältö laitto-
man/kyseenalaisen materiaalin varalta. App Store pyrkii tällä metodilla tarjoa-
maan käyttäjille mahdollisimman laadukasta ja haittaohjelmavapaata sisältöä.
Sovelluksen testaaminen ei ole läheskään täydellistä tai kattavinta, mutta tes-
tausvaiheessa ilmenevä bugi tai kaatuminen aiheuttaa automaattisesti hylkää-
vän päätöksen. Luonnollisesti sama koskee myös laitonta ja/tai kyseenalaista
sisältöä. (App Store Review Guideline, 2010.)

App Storen käyttöehdot ovat samalla linjalla Android Marketin kanssa esimer-
kiksi laittomasta sisällöstä, mutta ne sisältävät lisäksi myös huomattavan mää-
rän muita säännöksiä. Lisäksi useat näistä säännöksistä ovat tulkinnanvarai-
sia. Sovellus ei esimerkiksi saa olla nopeasti ja huolimattomasti tehdyn näköi-
nen ja sillä pitää olla *selkeä tarkoitus* tai se voidaan hylätä. Se ei saa näyttää
samalta kuin vakiona mukana tullut sovellus (esim. tekstiviestit, sähköposti,
musiikkisoitin jne.). Sama koskee myös ei-vakiona mukana tulleita sovelluksia.
Ohjelmat eivät saa muistuttaa liikaa toisiaan. Tällä pyritään rajoittamaan esi-
merkiksi samanlaisten pelien (pasianssi jne.) määrää. Ehkäpä kaikista oleelli-
sin ja samalla epätarkin linjaus löytyy kohdasta, jossa määritellään, että sovel-
lukset, jotka eivät ole kovin hyödyllisiä tai tarjoa viihdykettä pitkäksi aikaa voi-
daan hylätä. (App Store Review Guideline, 2010.)

Säännöt ovat erittäin tulkinnanalaisia, ja lopullinen päätös on aina Applen puo-
lilla. Useat kehittäjät ovatkin tätä vastaan. Tämä on saattanut olla yksi vaikut-
tajista, minkä vuoksi Android Market ei ottanut käyttöön tarkastusprosessia.
Toisaalta Applen pelisäännöt varmistavat ainakin jossain määrin sen, mitä on
luvattu, eli laadukkaan sisällön. Lisäksi jo pelkkä ohjelmien määrä kertoo sen,
että tavoitteena ei ole antaa hylkäyspäätöstä kaikille.

6 SOVELLUSKAUPPOJEN ANSAINTAMALLIT

Mobiilisovelluksilla ansaintaan on olemassa useita eri malleja, joista osa perustuu perinteiseen ohjelmistoteollisuuteen, osa Internet-palveluihin ja osa on täysin uusia mobiilimarkkinoita varten kehitettyjä. Ansaintamallia kannattaa miettiä jo sovellusta suunniteltaessa. Tässä luvussa on listattu suosituimpia menetelmiä, jotka pätevät useimpiin sovelluksiin. Pieniin markkinarakoihin keskittyvät mallit (kuten käyttäjätietojen myyminen) on jätetty pois. Mutta kuinka hyvin mobiilisovelluksilla voikaan ansaita ja mitä ovat keskiarvot, jos kehityskuluja ei huomioida? Siitä pyritään ottamaan seuraavassa selvää.

6.1 Tapaus Angry Birds

Angry Birds on suomalaisen Rovio-ohjelmistotalon kehittämä mobiilipeli, joka on saatavilla useimmille mobiilialustoille. Latauskertoja on kertynyt yhteensä yli 42 miljoonaa, joista 12 miljoonaa on maksullisia. Leijonanosa on ostettu Applen App Storesta. Rovio on onnistuneesti sekoittanut eri ansaintamalleja alustojen välillä. Esimerkiksi Symbian³:lla ja iOS:lla varustetut laitteet voivat ostaa pelin normaalisti sovelluskaupasta. Tarjolla on myös ilmainen kokeiluversio. Kun käännöstä alettiin tehdä Androidille, muodostui ongelmaksi runsas laitekirjo ja versioiden pirstoutuminen laitteiden välillä. Testaaminen ja säätäminen mahdollisimman laajan yhteensopivuuden takaamiseksi veivät paljon aikaa. Rovio päätti, että peli tulisi olemaan ilmainen ja se tuettaisiin mainoksilla (AdMob). Tällä hetkellä Android-version arvioidaan tuottavan noin miljoona dollaria kuukaudessa mainostuloja. (Parr 2010.)

Alkuperäisen julkaisun jälkeen peli on saanut säännöllisesti lisää sisältöä uusilla kentillä ja hahmoilla. Suurin osa on tarjolla ilmaiseksi, mutta myös maksullista sisältöä on saatavilla. Sittemmin suosion kasvaessa AngryBirds on noussut brändiksi. Tarjolla on pehmoleluja, t-paitoja ja tuleva Mattelin kehittämä lautapeli. Rovio myös tutkii aktiivisesti TV- ja elokuvamahdollisuuksia. Lisäksi pelin saatavuus on laajennettu mobiililaitteista lähes kaikkialle. Se on saatavilla PSP:lle ja PS3:lle PlayStation Mini -pelinä, Windowsille, Macille ja

käännökset ovat tulossa myös Xbox 360:lle, Nintendo Wii:lle ja Nintendo DS:lle. Tällainen menestystarina on tietysti erittäin poikkeuksellinen, mutta toimii hyvänä esimerkkinä sen mediassa saaman huomion vuoksi. Lisäksi kannattaa huomioida, että pelin alkuperäinen kehitystyö maksoi yli 100 000 euroa ja sen jälkeen rahaa on kulunut vielä moninkertainen määrä. (Parr 2010.)

6.2 Sovelluksen myynti

Myynti on tällä hetkellä yleisin käytössä oleva malli. Android Market ja App Store jakavat myynnistä saadut rahat samalla kaavalla: kehittäjä saa 70 % ja sovelluskauppa 30 %. Tämä suhde on yleisesti ottaen käytössä kaikkialla. Sovelluskaupan osuutta perustellaan kaupan ja palveluiden ylläpidon kustannuksilla. Poikkeuksena tästä on operaattorilaskutus, jolloin kehittäjän saama osuus vaihtelee. Pääsääntöisesti tässä metodissa se on kuitenkin vähemmän. Operaattorilaskutuksen saatavuus ja hinta vaihtelee kauppojen ja maitten välillä. Myynti tapahtuu pääsääntöisesti yksittäisinä tapahtumina kuluttajille, mutta massamyynti esimerkiksi yrityksille tai kouluille on myös mahdollista. Usein ohjelmasta on tarjolla ilmainen ja rajoittunut ns. lite-versio, jonka avulla kuluttajat voivat kokeilla sovellusta ilman ostopäätöstä. Lisäksi määritetyn hinnan päälle tulevat maakohtaiset verot. App Store tilittää ALV:n puolestasi, mutta Android Marketissa se on kehittäjän vastuulla. Verotusasiat saattavat aiheuttaa päänvaivaa, etenkin kun kyseessä on kansainvälinen myynti. Niitä selvittäessä kannattaa kääntyä ammattilaisen puoleen. (Murphy 2009.)

Esimerkkilaskelmia

Keskimääräisten odotustuottojen laskeminen on vaikeaa, sillä sovelluskaupat julkaisevat harvoin, tai eivät ollenkaan, virallista tietoa kehittäjille maksetuista summista. Ylipäätään tarkan tiedon saaminen latausmääristä, maksullisten latausten osuudet latausmääristä jne. on käytännössä mahdotonta. Tietyillä arvioilla voidaan kuitenkin tehdä joitakin suuntaa-antavia yleistyksiä.

Tutkimusyhtiö IHS Screen Digest arvioi (Dilger 2011), että App Storen tulot vuonna 2010 olivat noin 1,782 miljardia Yhdysvaltain dollaria. Kun tästä vähennetään Applen 30 %:n osuus, muodostuu kehittäjille maksetuksi summaksi 1,247 miljardia. Sovelluksia oli vuoden 2010 lopussa saatavilla noin 317 000. Sovelluskauppojen analysointiin erikoistunut Distimo arvioi (2010), että 73 % kaikista App Storen sovelluksista on maksullisia, jolloin kappalemääräksi muodostuisi 231 410. Näistä voidaan laskea, että yksi sovellus tienaisi keskimäärin noin 5388 dollaria vuodessa.

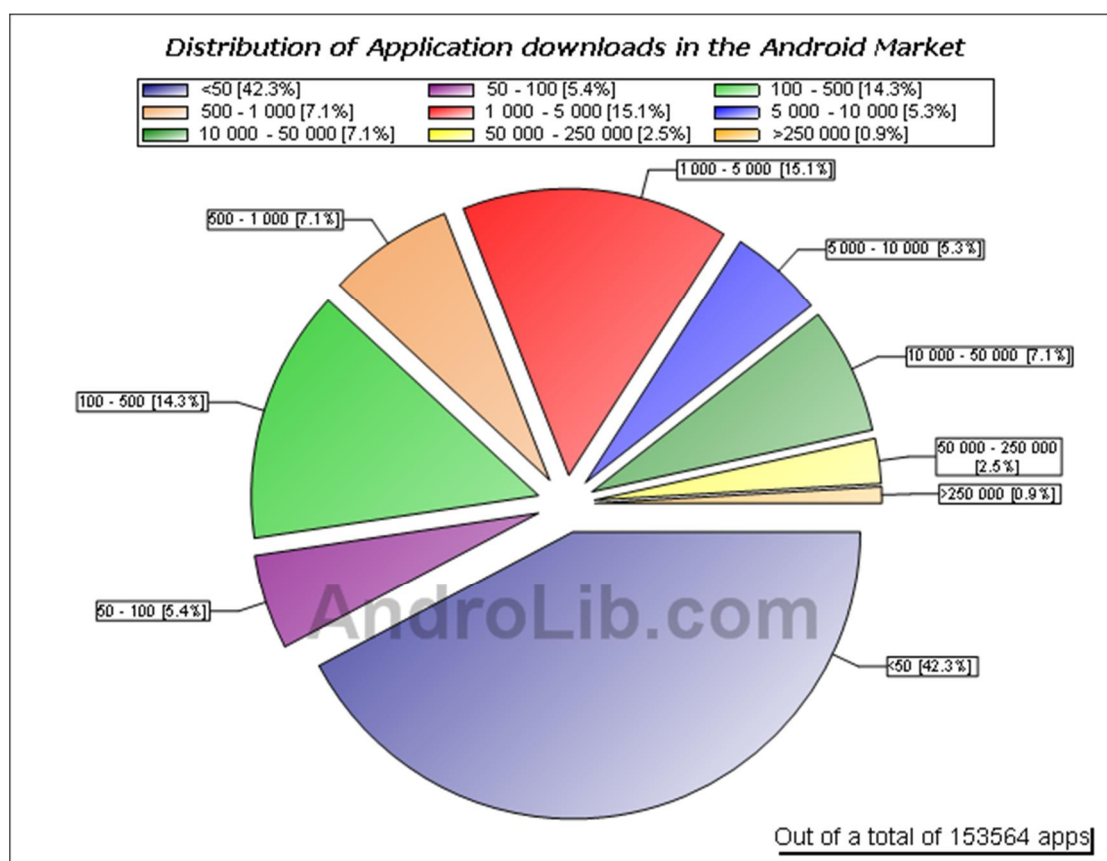
Lasketaan keskiarvo vertailun vuoksi myös toisen lähteen numeroilla. Heinäkuussa 2010, kaksi vuotta julkaisun jälkeen, Apple julkisti maksaneensa kehittäjille yli miljardi dollaria (tästä luvusta on vähennetty Applen 30 %:n osuus) kun kokonaislatauksia oli kertynyt noin viisi (5) miljardia. (Haselton 2010) Kymmenen miljardia latausta tuli täyteen tammikuussa 2011. Jos oletetaan, että sovelluksista maksettu keskihinta ja maksullisten latausten osuus on pysynyt samana, voidaan karkeasti arvioida kehittäjille maksetun summan kaksinkertaistuneen. Kymmenen miljardin latauksen kohdalla sovelluksia oli saatavilla noin 320 000, joista maksullisten osuudeksi muodostuu 233600. Näillä oletuksilla yksi ohjelma tienaisi kahden miljardin kakusta 2,5 vuoden aikana noin 8560 dollaria eli 3424 vuodessa.

Android Marketin kohdalla tilanne on paljon huonompi. Vaikka latauksia kertyykin huomattavia määriä, Google on julkisesti harmitellut maksullisten latausten vähäistä määrää. IHS Screen Digest arvioi Android Marketin vuoden 2010 tuloiksi 105 miljoonaa dollaria. Vähennetään tästä kaupan osuus 30 % ja kehittäjien kakuksi jää 73,5 miljoonaa. Tarjolla olevia sovelluksia oli vuoden lopussa noin 200 000. Vain 33,2 % näistä on maksullisia, jolloin lukumääräksi kertyy 66 400. Keskiarvoksi muodostuu 1107 dollaria.

Haluan painottaa, että nämä laskelmat eivät ole tarkkaa tiedettä. Luvut perustuvat yleistyksiin ja arvioihin, joten niitä voidaan pitää korkeintaan suuntaantavana. Latausten määrät ovat epävirallista tietoa ja niiden määrissä ei ole huomioitu inaktiivisia sovelluksia, joita pelkästään App Storessa arvioidaan olevan kymmeniä tuhansia. Nekin ovat todennäköisesti elinaikanaan jotain

tuottaneet. Lisäksi keskiarvoa vääristävät huomattavasti ääripäät eli miljoonia myyvät sovellukset.

Keskiarvoa tärkeämpi käsite on mediaani. Se kuvaa jakauman tyypillistä arvoa luotettavammin, koska sitä ei vääristä jakauman ääripäät. Pinch Media arvioi, että puolet kaikista App Storen ostetuista sovelluksista saa alle 1000 latausta elinikänsä aikana. (Northcott 2010.) Oletetaan tämän perusteella, että medianilukuna on tasan tuhat. Tätä kirjoitettaessa sovelluksen keskihinta on 2,50 dollaria. Tällöin tyypillinen sovellus siis tuottaa noin 2500 dollaria koko elinikänsä aikana. Ja oleellista tässä luvussa on se, että puolet kaikista ohjelmista tuottaa vielä vähemmän. Android Marketin kohdalla ei löytynyt tietoa maksullisten ohjelmien latausmääriä tai keskihinnasta, mutta kuviossa 6 nähdään jakauma kaikkien latausten kesken.



KUVIO 6. Android Marketin latausten jakautuminen. (Androlib 2010.)

Tietoa, että ilmaisia sovelluksia ladataan paljon enemmän kuin maksullisia, ei voitane pitää yllättävänä. Mutta kuinka iso ero onkaan? Jos kehittäjille on

maksettu yhteensä yli kaksi miljardia dollaria, tarkoittaa se, että App Store on yhteensä tienannut noin 2,86 miljardia dollaria. Kun tämä luku jaetaan tämän päivän (10.2.2011) sovelluksen keskihinnalla (2,50), saadaan maksettujen sovellusten määräksi 1,144 miljardia. Se puolestaan edustaa kaikista latauksista 11,4 %:n osuutta ja tämän perusteella loput 88,6 % kaikista App Storen latauksista olisi ilmaisia. Distimo tutki 300 suosituinta ilmaista ja maksullista ohjelmaa App Storessa. Ilmaiset ohjelmat keräsivät joka päivä joulukuussa 2010 yli kolme miljoonaa latausta yhteensä. Vastaavasti maksullisia ladattiin yhteensä vain 350 000 päivässä. Tämä tarkoittaa sitä, että 300 suosituinta maksullista ohjelmaa keräsivät keskimäärin 1167 latausta päivässä. Kun se kerrotaan keskihinnalla, tulokseksi saadaan 2917,5 dollaria. Tässäkin medianiluku kertoisi paremman totuuden, sillä esimerkiksi kymmenen suosituinta sovellusta keräävät taatusti myös kymmeniä kertoja enemmän latauksia kuin vaikkapa sovellukset sijalla 200–210. (Distimo 2010.)

Apple on ilmoittanut myyneensä yhteensä yli 160 miljoonaa iOS:illä varustettua laitetta. Jaetaan kymmenen miljardin latausmäärä tällä luvulla ja tuloksena on ladattujen sovellusten keskimäärä per laite eli 63. Vuositasolla keskimääräksi tulee 25. Tästä voidaan laskea, että keskimäärin iOS-laitteen käyttäjä ostaa vuodessa noin 3 sovellusta eli kuluttaa 7,5 dollaria.

Hinnoittelu

Vaikka suurin osa Top-listojen suosituimmista sovelluksista onkin hinnoiteltu alle Yhdysvaltain dollarin (tai euron), ei se automaattisesti tarkoita, että kyseessä on *paras* mahdollinen hinta. Ensin kannattaa tarkastella lähimmät kilpailijat ja tutkia heidän asettaman hinnoittelun vaikutusta. Ei ole olemassa mitään vakuuttavia todisteita siitä, että 0,99 dollarin sovellukset myisivät lopulta enemmän kuin ne jotka on hinnoiteltu 1,99 dollariin. Jos kyseessä on täysin uniikki sovellus, joka koskettaa selvästi pientä markkinarakoa (kuten liikemaa- ilma, tiedeyhteisöt, jne.), voi sen hinnoitella korkeammaksi. On hyvin todennäköistä, että heillä on myös varaa maksaa siitä. Mutta jos puhutaan ”suuren” yleisön sovelluksista, yli neljän tai viiden dollarin hintalapuilla sen pitää myös olla jotain todella erikoista. Viisainta lienee pitää hinta alle kolmen dollarin,

jolloin sen todennäköisyys päätyä impulssiostokseksi pysyy korkeana. Vaikka sovellus olisikin jotain todella erikoista ja täten ansaitsisi ison hintalapun, kilpailijoita ilmestyy nopeasti ja heidän ensimmäinen ässänsä hihassa on myydä halvemmalla. (Wooldridge ym. 2010, 304-305.)

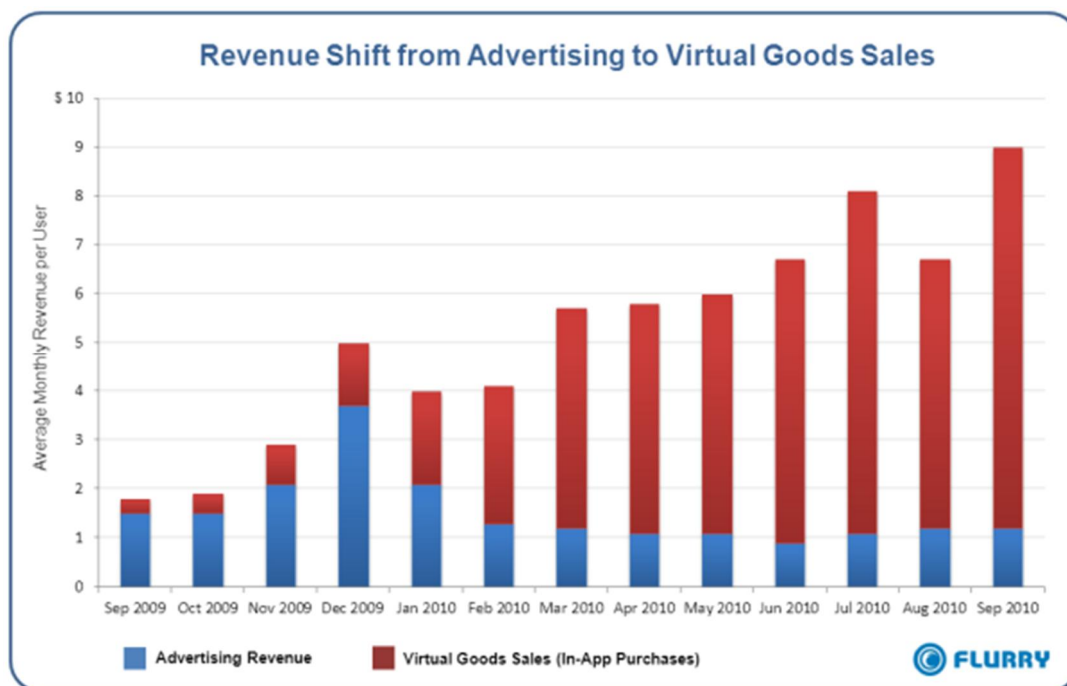
Monet pitävät hinnan matalana toiveenaan päästä sen avulla mahdollisimman korkealle ostetuimpien sovellusten joukossa, mikä tietysti lisää myyntiä huomattavasti. Mutta mitä sitten, kun myynnin piikki on tapahtunut ja kiinnostus alkaa laskea? Vain harvat ohjelmat pysyvät suosituimpien joukossa viikosta tai kuukaudesta toiseen. Lopputulos ratkaisee, eikä hetkellinen suosio. On hyvin todennäköistä, että myyntiluvut päätyvät samaan, maksoi sovellus sitten 0,99 tai 1,99 dollaria. Erona tässä on, että kalliimman hinnan valinnut on tienannut puolet enemmän. Lisäksi korkeampi hinta mahdollistaa myöhemmin alennusmyynnin, joka voi tuoda uutta potkua myyntilukemiin. (Wooldridge ym. 2010, 304-305.)

6.3 Freemium

Sana freemium tulee englannin kielen sanoista "Free" (ilmainen) ja "Premium" (tässä kontekstissa lisämaksu). Freemium on siis yhdistelmä eri malleja ja strategioita. Siinä ideana on antaa osa palvelusta ilmaiseksi ja veloittaa maksua lisätoiminnoista, sisällöstä tai vastaavasta. Ja kuten edellisestä kappaleesta opimme, ilmaiset sovellukset keräävät paljon enemmän huomiota kuin maksulliset. Freemium tarjoaa hurjan määrän mahdollisuuksia, jossa kliseen mukaisesti vain mielikuvitus on rajana. Esimerkiksi kehittäjä voi tarjota luomaansa peliä ilmaiseksi, mutta vain maksullisessa versiossa on mukana moninpeli. Vaihtoehtoisesti siihen on mahdollista ostaa lisää sisältöä kuten uusia kenttiä. Musiikkiaiheisiin sovelluksiin voi ostaa lisää kappaleita jne. Jos sovellus on suunnattu pääasiassa yrityksille, voi kehittäjä veloittaa erikseen vaikkapa käyttötuesta. (Murphy 2009.)

Freemium kasvattaa koko ajan suosiotaan ja siinä käytetään pääasiassa hyväksi alustojen mahdollisuutta ostaa lisäsisältöä suoraan ohjelmasta. Käyttäjät

voi helposti nappia klikkaamalla suorittaa oston sulkematta sovellusta. Voitonjako on yleensä sama kuin sovellusten myynissä eli 70 % kehittäjälle ja loput kaupalle. Tällä hetkellä ainakin iOS ja BlackBerry OS mahdollistavat ohjelmis- sa sisäisen oston ja Android on saamassa ominaisuuden keväällä 2011.

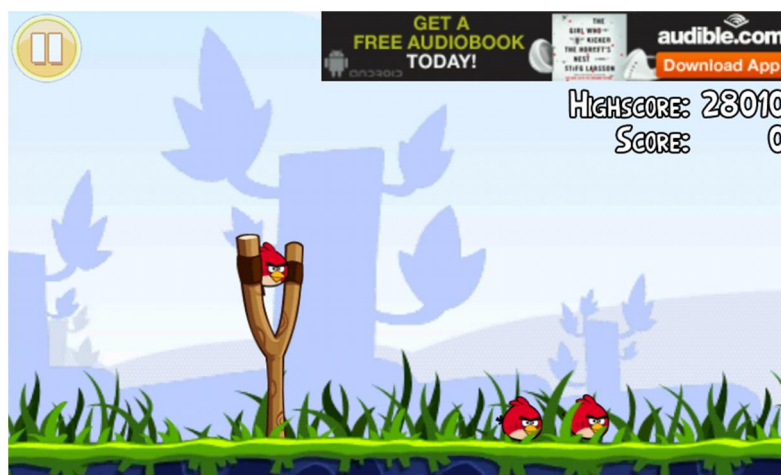


KUVIO 7. Lisäsisällön myynnin kehittyminen mainostuloihin verrattuna. (Flurry 2010)

Kuviossa 7 nähdään, kuinka paljon lisäsisällön myynnin kannattavuus on noussut verrattuna mainostuloihin. Syyskuussa 2010 yksi käyttäjä tuotti noin yhden dollarin verran mainostuloja ja lähes 8 dollaria sisällön myynnillä. Data on kerätty johtavista sosiaalisten median ohjelmista ja peleistä, joilla on yhteensä yli 2.2 miljoonaa päivittäistä käyttäjää. On huomioitava, että data on kerätty ainoastaan iOS-laitteista ja ohjelmien otanta edustaa vain pienehköä määrää kokonaiskuvasta. Tästä voidaan kuitenkin päätellä, että mallissa on rahaa liikkeellä. Lisäksi täytyy ottaa huomioon, että sisällön myynti ja mainosten käyttö eivät sulje toisiaan pois.

6.4 Mainokset

Mainosten myynti voidaan laskea osaksi Freemium-mallia eli ohjelma itsessään on ilmainen, mutta se on mainoksilla tuettu. Tietysti kukaan ei estä mainosten sisällyttämistä myös maksullisiin ohjelmiin, mutta siinä tapauksessa kannattaa varautua negatiiviseen palautteeseen asiakkailta. Mainos on tyypillisesti banneri (kuvio 8), joka ilmestyy sovelluksen käytön aikana ruudulle. Mainoksen sijoittaminen on kehittäjästä kiinni, sillä se voi ilmestyä esimerkiksi pelin alitusvalikossa tai kesken pelin ja olla ruudulla hetken aikaa tai pysyvästi. Mobiilimainonnan yksi keskeisin ongelma ovat virheklikkaukset. Näytöt ovat suhteellisen pieniä, jolloin käyttäjät klikkaavat mainoksia vahingossa. Tämä aiheuttaa usein ärsytystä. (Murphy 2009.)



KUVIO 8. Mainoksia AngryBirds pelin Android-versiossa

Mainosten toiminta perustuu eri mobiililaitteille suunnattuihin mainosverkostoihin. Suosituin ja markkinajohtaja näistä on nykyisin Googlen omistama AdMob. Sen avulla on mahdollista mainostaa mm. Android- ja iOS-alustoissa. Applen vastaava palvelu on iAd, mutta se toimii vain iOS-tuoteperheessä. Lisäksi markkinoilla on useita kolmannen osapuolen verkostoja. Tulosta syntyy yleensä joko mainoksen klikkausten tai näyttökertojen perusteella, riippuen mainosverkosta. Verkot eivät tarjoa kiinteitä hintoja, sillä tulos riippuu mainoksesta, mainostajasta, sijainnista, kohderyhmästä ja kymmenistä muista tekijöistä. Myös tulojen jakautuminen vaihtelee palveluiden välillä, mutta alan tyypillinen jakauma on 60/40 kehittäjän hyväksi.

Keskimääräisten mainostulojen laskeminen on erittäin hankalaa. Parhaiten ansaintamahdollisuudet selviävät vain kokeilemalla. Mainosverkon valitsemisessä kannattaa etsiä esimerkkejä kehittäjien tuloksista Internetistä. Vertailua tehdään muutamilla termeillä.

- Requests eli mainoksen pyyntökerrat
- Impressions eli mainoksen näyttökerrat
- Fillrate eli pyyntökerroista seuranneet näyttökerrat, täyttöaste
- CTR (Click-through-rate) eli klikkausprosentti näyttökerroista
- CPC (Cost-per-click) eli paljonko yhden mainoksen klikkaus maksaa
- eCPM (Effectivecost per mille) eli paljonko tuhat näyttökertaa maksaa tai tuloja laskettaessa paljonko tuhat näyttökertaa tuottaa.

Pyyntökerta syntyy joka kerta, kun joku käyttää sovellusta ja sovellus pyytää mainosta palvelimelta tai palvelin pyrkii näyttämään mainoksen. Näyttökertaa ei synny, jos käyttäjä ehtii siirtyä kohdasta pois tai palvelimella ei ole sopivaa mainosta. Tämä puolestaan johtuu siitä, että mobiilimainokset ovat hyvin optimoituja. On tärkeää huomata, että näyttökerrat eivät ole sama asia kuin käyttäjien määrä. Koska mainokset vaihtuvat tietyn väliajoin, voi yksi käyttäjä nähdä sovelluksen käytön aikana useita mainoksia. Täyttöasteen merkitys on suuri, sillä mitä isompi se on, sitä enemmän on mahdollista ansaita. Esimerkiksi iAD:in täyttöaste on vielä alhainen, koska se on suhteellisen uusi tekijä markkinoilla ja toimii vain muutamassa maassa. Admobilla on puolestaan markkinajohtajana yleensä isoin täyttöaste. Toisaalta iAD veloittaa mainostajilta huomattavasti korkeampia maksuja kuin Admob ja tämä näkyy myös jaettavissa tuloissa. (Campbell n.d.)

Sovelluksen suosio vaikuttaa tuloihin ratkaisevasti. Eikä pelkästään latausten määrä vaan se kuinka paljon sitä käytetään päivittäin. Angry Birds tienaa mainoksilla paljon rahaa, koska käyttäjät pelaavat sitä päivittäin ja se on demograafiselta luonteeltaan universaali. Jos sovellus on ns. kertakäyttöinen, jäävät mainostulot pitkällä aikavälillä pieniksi. Localyticsin tutkimuksen mukaan 26 % käyttää ohjelmaa vain kerran latauksen jälkeen. Mainoksilla tuetussa ohjelmassa tämä ei ole hyvä uutinen. (Localytics 2011.)

Käytetään esimerkkinä Crazy Snowboard peliä. Se on maksullinen App Storessa, mutta Android-versio on ilmainen ja AdMobin mainoksilla tuettu. Pelin kehittäjä ilmoitti 30 päivän tulot, kun latauksia oli reilut 385 000 (sittemmin pelin suosio on moninkertaistunut). Summat ovat Yhdysvaltain dollareina.

Tulot	eCPM	Pyyntökerrat	Näyttökerrat	Täyttöaste	CTR
2066,99 \$	0,29 \$	7 457 152	7 142 018	95,77 %	0,61 %

Kuten todettiin, tulojen vertaaminen tai suhteuttaminen on hankalaa tiedon puuttuvuuden vuoksi. Vaikka vertauskuvana olisi toinen peli, jolla on sama latausmäärä, eivät luvut todennäköisesti täsmäisi. Muuttuvia tekijöitä on liikaa. Mikäli kuvio 6 pitää paikkansa, kyseinen ohjelma kuuluu siihen marginaalivähemmistöön (alle 1 %), jotka keräävät yli 250 000 latausta. Tämä esimerkki kuitenkin todistaa, kuinka tärkeää on käyttäjien iso määrä ainakin kyseisessä mainosverkossa. Vain reilusti alle 1 % mainoksen näkijöistä klikkasi sitä. Tuhat näyttökertaa tuotti 0,29 dollaria. Vaihtoehtoisesti lisää perspektiiviä asiaan saadaan myös seuraavasti: yksi käyttäjä tuotti 30 päivän aikana kehittäjälle noin puoli senttiä. (Edis 2011.)

6.5 Kumppanuusmarkkinointi

Kumppanuusmarkkinointi (Affiliate Marketing) on markkinointimuoto, jossa kehittäjä saa korvauksen tuloksen tuottamisesta kumppaneilleen. Tämä tapahtuu jonkin kumppanuusohjelman avulla. Esimerkiksi Apple tarjoaa omaa ohjelmaansa ”iTunes & App Store Affiliate Program”, jonka rajapinnan avulla sovellukseen voidaan sisällyttää linkki iTunes-verkkokauppaan. Jos sovelluksen sisältä tullut klikkaus johtaa tuotteen ostoon iTunesissa, saa kehittäjä siitä provision. Kyseisessä verkkokaupassa on mobiilisovellusten lisäksi tarjolla musiikkia, elokuvia, tv-sarjoja jne. Kumppanuusmarkkinointia ei voida sisällyttää jokaiseen sovellukseen, sen täytyy jotenkin olennaisesti liittyä kaupattavaan tavaraan. Esimerkiksi Apple tarkastaa jokaisen kumppaniksi pyrkivän. Lisäksi kumppanuus voi rajoittua tekijänoikeuksien vuoksi tiettyihin maihin tai alueisiin. (Murphy 2009.)

Lasketaan karkea esimerkkituotto. Oletetaan, että sovellus liittyy jollain tavalla vaikkapa musiikkiteollisuuteen. Tarkoituksena on saada käyttäjät ostamaan ohjelman kautta musiikkia. Applen kumppanuusohjelma maksaa provisiota 4-5 % myynnistä. Euroopan alueella iTunesin alin mahdollinen hinta yksittäiselle kappaleelle on 0,69 euroa ja korkein 1,29 euroa. Tällöin keskihinnaksi saadaan 0,99 euroa. Jos provisio on 5 %, tällöin tuotto yhdestä 99 sentin kappaleesta on 4,95 senttiä. Tuhannen euron tiliin pitää saada käyttäjiä ostamaan ohjelman kautta 20 202 kappaletta. Nyt täytyy huomioida, että suurin osa käyttäjistä ei välttämättä tule edes koskaan klikkaamaan kumppanuuslinkkiä. Ja klikanneistakin vain pieni yleensä johtaa ostoon. Jos oletetaan, että 20 % klikkauksista johtaa ostoon, pitäisi tuhannen euron tiliä varten saada ensin yli 101000 potentiaalisesti kiinnostunutta käyttäjää. Kumppanuusmarkkinointi toiminee kaikista parhaiten jonkin toisen mallin tukena. Lisäksi edellytyksenä on hyvä ja koukuttava kumppanuusmarkkinointiin sopiva idea. (Apple Affiliate Program, n.d.)

6.6 Tilausmalli

Tilausmalli on yleisimmin käytössä sanoma- ja aikakauslehdillä. Esimerkiksi Helsingin Sanomat tarjoaa iPhoneille ja iPadille sovellusta, jolla voi lukea painetun lehden digiversiota laitteessa. Sovellus on ilmainen, mutta digilehden lukeminen vaatii tilauksen tai lukuajan ostamista. Tekniikan Maailma puolestaan myy lehteä irtonumeroina App Storessa. Toistaiseksi tilausmallia ei ole saatavilla kaikilla alustoilla. Lisäksi kannattaa huomioida App Storen käyttöehdot. Jos sovellus perustuu esim. vuositilaukseen ja sen voi tehdä sovelluksen Internet-sivuilta, täytyy se olla myös mahdollista App Storen kautta. Tällöin Apple ottaa tilaushinnasta oman 30 %: osuutensa. (Murphy 2009.)

7 JULKAISUN JÄLKEEN

Markkinointi saattaa nousta monella vielä isommaksi haasteeksi, kuin varsinainen sovelluksen luominen. Useat kehittäjät sanovatkin, että varsinainen työ alkaa vasta sovelluksen julkaisun jälkeen. Ainakin siinä tapauksessa, että aikomuksena on ansaita rahaa. Monet menestystarinat, joissa itsenäinen kehittäjä on tienannut kymmeniä tai satoja tuhansia euroja muutamassa viikossa, sijoittuvat App Storen alkutaipaleelle. Heidän etunaan oli olla ensimmäisten joukossa. Tänä päivänä tilanne on toinen. Android Marketissa on yli 250 000 ja App Storessa yli 350 000 sovellusta tarjolla. Miksi käyttäjät valitsivat juuri sinun ohjelmasi? Uniikki, alkuperäinen idea ja hyvä käytettävyys on tietysti ideaalinen lähtökohta, mutta ei takaa menestystä. Pahin virhe on luulla, että sovellus saa latauksia kuin itsestään.

Kuviosta 5 voidaan todeta, että 60 % iPhoneen käyttäjistä löytää ohjelmansa AppStoren Top-listoja selaamalla. Voidaan olettaa, että sama pätee myös muihin sovelluskauppoihin, sillä kaikkien Top-listat ovat hyvin saman tyyppisiä. Ensimmäisenä käyttäjille avautuu kaikista suosituimmat maksulliset ja ilmaiset ohjelmat sekä ”Featured” -osio, jossa on sovelluskaupan henkilökunnan valintoja. Lisäksi suosituimpia sovelluksia voidaan selata kategorioittain. Top-listalle pääseminen on käytännössä ehdotonta menestymisen kannalta. Siellä olevat sovellukset saavat moninkertaisesti enemmän näkyvyyttä ja latauksia. Sama pätee Featured -listaa, mutta siihen pääsyyn ei voi vaikuttaa itse. Todennäköisyys räjähtävään suosioon ei ole kehittäjän puolella. Mainosyhtiö W3i arvioi, että pääsy AppStoren Top 25 ilmaiset ohjelmat -listalle vaatii yli 50 000 päivittäistä latausta pelkästään Yhdysvalloissa. (W3i. 2011.)

Suosio on monen eri tekijän yhteissumma, jossa isoa osuutta edustaa puhtaasti onni. Mobiilisovelluksia tehtailevalla yrityksellä on usein käytössään oma markkinointiosasto, joka hoitaa ammattimaisen promotioon. Mutta entäpä yksityinen käyttäjä, jolla ei ole resursseja? Oikeaa markkinoinnin menestyskaavaa ei ole olemassakaan, mutta seuraavassa käydään läpi muutamia perusasioita.

7.1 Analyysityökalut

Sovelluskaupat tarjoavat hyvin alkeellisia työkaluja myynnin ja latauksien analysointiin. Hyödyntämällä kolmannen osapuolen työkaluja on mahdollista saada selville, *kuinka* sovellusta käytetään sekä paljon muuta hyödyllistä. Analytiikan avulla on mahdollista selvittää esimerkiksi, millä käyttöjärjestelmän versioilla sovellusta käytetään, kuinka paljon aikaa käyttäjät viettävät sen parissa ja mitkä ominaisuudet/toiminnot ovat suosituimpia. Lisäksi on mahdollista saada tietoa demografiasta, mikä edesauttaa edelleen kehitystä ja markkinointia. Useat työkalut kykenevät myös tuottamaan huomattavasti monipuolisempaa dataa lataus- ja ostomääristä kuin sovelluskauppojen omat vastaavat. Jos esimerkiksi kyseessä on peli, josta on tarjolla ilmainen "lite-versio" ja maksullinen kokoversio, näiden työkalujen avulla on mahdollista saada selville, kuinka moni ilmaisen version käyttäjästä päätyy ostamaan kokoversion. Vakio-työkaluilla tämä ja moni muu asia ei ole mahdollista. Suosittuja palveluita ovat Flurry, TapMetrics ja Heartbeat. Google-ilmoitukset on palvelu, joka ilmoittaa sähköpostilla, kun valitusta (esim. sovelluksen nimi) aiheesta puhutaan verkossa. Tämä käsittää hakusanat, uutiset, blogit jne. (Yarmosh 2010, 208.)

7.2 WWW-sivut, sosiaalinen media

WWW-sivujen päätarkoituksena ei ole markkinoida tai muutoin tuoda näkyvyyttä sovellukselle. Tämä ihan siitä syystä, että sivusto hukkuu jo luomisvaiheessa Internetin syöveriin. Sen sijaan sivuston tarkoituksena on palvella sovelluksen käyttäjää ja sitä kannattaa mainostaa sovelluskaupan esittelysivulla. Mukana voi olla uutisia tulevista päivityksistä, videoita ja kaikkea muuta asiaan liittyvää. Luonnollisesti sen tulisi sisältää ohjelman esittely ja yhteystiedot. Kannattaa myös tarkastaa, että sivut toimivat mobiiliselaimella.

Sosiaalinen media on nykypäivänä käytännössä pakollinen juttu johon on hypättävä mukaan. Se on tullut jäädäkseen, halusi sitä tai ei. Se tarjoaa vaivattomasti ilmaista markkinointia ja lähes täydellisen kanavan asiakaspalveluun

sekä muuhun vuorovaikutukseen käyttäjien kanssa. Facebook lienee jo kaikille tuttu käsite 600 miljoonan käyttäjän voimallaan. Otetaan esimerkiksi sen julkaisema WWW-sivuille liitettävä liitännäinen, jossa sisällöstä voi ”tykätä” tai vaihtoehtoisesti suositella sitä. Nappia painamalla kyseinen tapahtuma ilmestyy kaikille ystäville Facebookiin. Liitännäinen on julkaisun jälkeen levinnyt lähes joka paikkaan. Sen käyttö on jo niin arkipäivää useilla, ettei sitä edes tule ajatelleeksi. Tiedon jakaminen onkin sosiaalisen median suosion takana. Ihmiset ovat kiinnostuneita toisten tekemisistä.

Kuviosta 5 ilmenee, kuinka tärkeitä suositukset ovat sovellusten löytämisen kannalta. Sen vuoksi onkin oleellista ylläpitää profiilia ainakin Facebookissa ja Twitterissä sekä promota niitä aktiivisesti. Suomessa Twitterin käyttö ei ole vielä oikein ottanut tuulta purjeisiinsa, mutta muualla maailmassa se on suurinta huutoa. Twitter on Top 10 -listalla maailman suosituimmista sivustoista. Ennen kaikkea se on iso juttu Yhdysvalloissa, joka sattumoisin on myös yksi suurimpia mobiilisovellusten markkina-alueita. On totta, että iso osa siellä tapahtuvasta keskustelusta voidaan luokitella tyhjänpäiväiseksi jaaritteluksi, mutta toisaalta sen osuutta Internetissä ei voi vähätellä. Jälleen kerran kannattaa tutustua kilpailijoihin. Mitä he tekevät siellä ja millaisin tuloksin? Ei kannata pelästyä, vaikka Facebookissa ei heti olekaan sankka määrä faneja tai Twitterissä seuraajia. Pientäkin käyttäjäkuntaa kannattaa palvella hyvin. Vastapalveluksena he levittävät sanaa eteenpäin. Lumipallo saadaan liikkeelle, kun sata ihmistä suosittelee sovellusta eteenpäin ja he suosittelevat sitä eteenpäin jne. (Yarmosh 2010, 236.)

7.3 Päivitykset

Sovelluskehityksen elinkaari ei pääty julkaisun jälkeen, pikemminkin päinvastoin. Sovelluksen ensimmäinen versio on harvoin täydellinen. Vaikka siinä ei ilmenisikään ohjelmointivirheitä tai kaatumisia, kannattaa päivitysten avulla hioa käytettävyyttä ja tuoda lisäarvoa. Jos virheitä tulee ilmi, tulisi niihin tarttua mahdollisimman pian ja antaa käyttäjille tieto, että korjaus on tulossa. Päivitysten puute saattaa aiheuttaa negatiivista palautetta ja karkottaa uusia asiakkai-

ta, jotka luulevat, ettei sovellusta enää tueta. Uusien asiakkaiden haaliminen on tietysti tärkeää, mutta yhtä tärkeää on pitää nykyiset käyttäjät tyytyväisinä. Toki kaikkien sovellusten elinkaari tulee aikanaan tiensä päähän, mutta päivittämisen tulisi jatkua mahdollisimman pitkälle. Etenkin mainoksilla tuetun sovelluksen elinehto on pitkä käyttöikä. Niin kauan kun sovellusta käytetään aktiivisesti ja käyttäjät ovat tyytyväisiä, he myös puhuvat siitä ja levittävät sanaa eteenpäin. Lisäksi tyytyväiset asiakkaat tukevat todennäköisemmin myös muita tuotoksiasi. (Wooldridge ym 2010, 369.)

7.4 Asiakaspalaute

Sovelluskaupat sisältävät oman palautekanavan, jossa käyttäjät voivat antaa sovellukselle arvosanan ja jättää kommentit. Sovelluksen esittelysivulla näkyy keskiarvo ja muutama uusin kommentti. Jos ohjelma saa jatkuvasti vain yhtä tähteä ja kommentit ovat pääsääntöisesti negatiivisia, karkottaa se helposti potentiaaliset uudet asiakkaat. Vastaavasti hyvä palaute auttaa asiakkaiden saannissa. Ensimmäinen johtopäätös jatkuvasta negatiivisesta palautteesta on, että ohjelma on yksinkertaisesti huono tai se ei toimi kunnolla. Tietysti tämä usein myös pitää paikkansa, mutta syyt voivat löytyä muualtakin. Täytyy muistaa, että palautteen antaminen on anonyymia. Tällöin kynnyks antaa laadutonta (törkeydet, lapsellisuudet yms.) palautetta on huomattavasti pienempi. Tälle ei yksinkertaisesti voi mitään. Huomionarvoisinta palautetta ovat vikailmoitukset, rakentava kritiikki ja muut parannusehdotukset. Niihin nopeasti tarttuminen ja hyödyntäminen voi kääntää latausmäärät parempaan suuntaan.

Toinen tapa saada palautetta on suoraan sähköpostitse tai sosiaalisen median kautta. Toisin kuin sovelluskauppojen palautteeseen, sähköposteihin ja Twitterin/Facebookin kautta saatuihin viesteihin tulisi vastata henkilökohtaisesti. Kukaan ei tietysti pakota näin tekemään, mutta asiakaspalvelun tärkeyttä ei voi olla painostamatta tarpeeksi. Palautteeseen vastaaminen on aikaa kuluttavaa hommaa, mutta se palkitsee lopulta. Mikä parasta, se on täysin ilmaista. Kun takana ei ole isoa markkinointikoneistoa, kaikki ilmainen on aina kotiin päin. Käyttäjät arvostavat sitä, että saavat tuntea olevansa mukana oh-

jelman kehityksessä. Tietysti on huomioitava, että kaikkien miellyttäminen on täysin mahdotonta, mutta yhdenkään mielen kääntäminen positiiviseksi auttaa. (Wooldridge ym 2010, 304 - 305.)

7.5 Ilmaisia promootiokeinoja

Mobiilisovellusten suosion noustessa mukaan tulivat myös niiden arvosteluihin ja palkintojen jakamiseen keskittyvät sivustot. Lisäksi niistä kirjoitetaan sanoma- ja aikakauslehdissä. Useat teknologiaan keskittyvät julkaisut luovat listoja tyyliin ”Parhaimmat sovellukset 2010” tai ”50 parasta mobiilipeliä juuri nyt”. Jos sattuu onni potkaisemaan (tai omaat yhteyksiä mediaan) ja joku tunnettu taho kehuu ohjelmaasi, siitä kannattaa tietysti ottaa täysi hyöty irti. Pelkkä maininta lehdessä/Internetissä lisää kiinnostusta. Kyseisestä arvostelusta kannattaa lisätä lyhyt lainaus tai kommentti sovelluskaupan kuvaelmaan ja mielellään heti ensimmäiselle riville. Samaa näkee esimerkiksi elokuvien promootiossa. Erilaisiin kilpailuihin osallistuminen on ilmaista, eikä siinä häviä mitään, vaikkei pääsisi edes karsinnoista läpi. Muutamia kilpailuja ovat esimerkiksi: Apple Design Awards, Best App Ever Awards ja AppFire’s App Star Awards. (Yarmosh 2010, 260.)

Useat, arvosteluihin keskittyvät sivustot myös vastaanottavat ehdotuksia, jota kannattaa hyödyntää. Kaikki eivät välttämättä lämpene jatkuvaan oman sovelluksen promoamiseen, mutta sitkeys usein palkitaan. Ja miksi rajoittaa vain perinteisiin sivuihin? Arvosteluja tehdään myös YouTube:ssa, jossa useilla kanavilla on jopa kymmeniä tuhansia seuraajia. AppStore mahdollistaa promootiokoodien luomisen kehittäjille, joiden avulla pystyy maksullisen sovelluksen lataamaan ilmaiseksi. Antamalla promootiokoodi vastineeksi arvostelusta kasvattaa mahdollisuuksia, koska tällöin arvostelijan ei tarvitse itse maksaa siitä. Promootiokodeja voi myös käyttää hyväksi vaikkapa järjestämällä jonkunlaisen kilpailun yhteistyössä arvostelusivuston (tai vastaavan) kanssa. Jos liittyy seuraajaksi Twitteriin tai alkaa faniksi Facebookissa, arvotaan heidän kesken koodeja. (Yarmosh 2010, 260.)

8 POHDINTA

Mobiilialustaa valittaessa ei kannata välttämättä tuijottaa pelkästään ohjelmointikieltä. Ohjelmointia suuremmaksi haasteeksi voi vielä nousta käyttöjärjestelmien versioiden pirstoutuminen sekä laitteiden määrä erilaisine spesifikaatioineen ainakin, jos tarkoituksena on saada oma sovellus käyttöön mahdollisimman laajalle. Kehityksen vauhdin hidastumista ei ole näköpiirissä. Mobiilikäyttöjärjestelmät päivittyvät useita kertoja vuodessa ja uusia laitteita ilmestyy käytännössä kuukausittain. Etenkin Androidin kohdalla jatkuva kehitys ja erilaiset laitteet saattavat olla lopulta jopa haitaksi. Sen markkinaosuus ei ole häviämässä minnekään, päinvastoin, mutta alustan pirstoutuneisuus saattaa etäännyttää joitakin kehittäjiä. Se ei ole hyvä asia, kun huomioidaan, että sovelluskaupan tarjonta vaikuttaa yhä useamman kuluttajan ostopäätökseen. Työssä tuli myös kumottua yleinen väite, että iOS-laitteille ohjelmointi on helppoa, koska malleja on vain yksi. Todellisuudessa myös iOS:n kohdalla joutuu miettimään eri laitteita ja tasapainoilemaan muutaman eri käyttöjärjestelmäversion kanssa. Toisaalta ongelma ei ole lähellekään yhtä haastava kuin Androidin kohdalla.

Mobiilisovellusten tekeminen ansaintatarkoituksessa ei ole automaattinen kultaivos. Toisaalta tämän ei pitäisi tulla kenellekään yllätyksenä. Työn tarkoitus ei ollut todistaa, ettei mobiilisovelluksilla voisi ansaita rahaa ja sitä ei myöskään tehty. Sen sijaan tuloksena on ainakin jossain määrin herätys todellisuuteen. Sovelluskaupat pursuavat tarjontaa ja todennäköisyys siellä menestymiseen on lopulta pieni. Toki tämäkin riippuu kuinka menestyminen määritellään. Jos lähtökohtana on ansaita hieman taskurahaa esimerkiksi opiskelujen ohella, todennäköisyys tavoitteen saavuttamiseksi on isompi. Jos sen sijaan mietitään sovellusten tehtailua puhtaasti elinkeinona, on tilanne hyvin epävarma. Kun mukaan pitää ottaa huomioon kehitys-, sijoitus- ja elinkustannukset, täytyy tuotosten olla jotain erikoista ja onnen minimointi maksimissa.

Työssä lasketut luvut ja johtopäätökset eivät ole kiveen kirjattuja. Kolmannen osapuolen data-arviot ovat kirjaimellisesti vain *arvioita*. Todellisuudessa keskimääräiset ansiot voivat olla paljonkin tässä tehtyjä johtopäätöksiä korkeam-

pia tai vastaavasti pienempiä. Mielestäni hyvä kehitysidea olisikin luoda sovellus ja koettaa työssä esiteltyjä ansaintamalleja esimerkiksi muutaman kuukauden ajan. Tutkittavia aihekokonaisuuksia on paljon tarjolla. Selvitettävänä voisi olla vaikkapa mainosverkkojen eroja tai se, onko pienemmällä hinnalla positiivista vaikutusta pitkässä juoksussa, kun puhutaan hinnoittelustrategioista. Yksi näkökulma olisi luoda sovellus usealle alustalle ja tehdä sen pohjalta kattava vertailupaketti ohjelmoinnista.

Loppujen lopuksi mobiilisovellusten liiketoiminta on kuin mikä tahansa muukin business. Sitä voidaan verrata esimerkiksi Suomen musiikkiteollisuuteen. Kaikista harrastajista vain pieni prosentti tienaa elantonsa musiikilla. Eniten näkyvillä ovat huippumenestyjät, jotka määrällisesti edustavat kaikista pienintä luokkaa. Heikin ovat todennäköisesti tehneet vuosia ja vuosia työtä menestysensä eteen. Vastaavasti heillä on myös saattanut käydä hyvä onni tai takana on valtava markkinointikoneisto (vrt. Idols, isot levy-yhtiöt). Mediassa ovat harvemmin esillä ne muusikot, jotka joutuvat hakemaan elantonsa päivätöistä, vaikka heidän musiikkinsa olisikin poikkeuksellisen hyvää. Opinnäytetyön tekijän mielestä tämä vertaus kuvaa hyvin mobiilimaailmaa. Kaikki eivät voi olla rokkitähtiä.

LÄHTEET

Android Developers. 2011. Androidille kehittäjien sivusto. Viitattu 10.1.2011.
[Http://developer.android.com/](http://developer.android.com/).

Apple Inc. 09.09.2010.App Store Guidelines.Viitattu 20.01.2011.
[Http://stadium.weblogsinc.com/engadget/files/app-store-guidelines.pdf](http://stadium.weblogsinc.com/engadget/files/app-store-guidelines.pdf).

App Inventor. 2011. About App Inventor. Viitattu 28.02.2011.
[Http://appinventor.googlelabs.com/about/](http://appinventor.googlelabs.com/about/).

Apple Developer - iOSDev Center. 2011. Viitattu 15.01.2011.
[Http://developer.apple.com/devcenter/ios/index.action](http://developer.apple.com/devcenter/ios/index.action).

Apple.n.d. App Store Affiliate Program. Viitattu 27.01.2011.
[Http://www.apple.com/itunes/affiliates/resources/documentation/app-store-affiliate-program.html](http://www.apple.com/itunes/affiliates/resources/documentation/app-store-affiliate-program.html).

BlackBerry. 2011. BlackBerry App World Vendor Supprot. Viitattu 01.03.2011.
[Http://us.blackberry.com/developers/appworld/](http://us.blackberry.com/developers/appworld/).

Campbell, J. n.d. Internet Marketing Terminology Explained. Viitattu 25.01.2011.
[Http://www.nightcats.com/samples/jargon.html](http://www.nightcats.com/samples/jargon.html).

Dilger, D. AppleInsider. 18.02.2011. IHS Screen Digest analysis. Viitattu 22.02.2011.
[Http://www.appleinsider.com/articles/11/02/18/rim_nokia_and_googles_android_battle_for_apples_ios_scraps_as_app_market_sales_grow_to_2_2_billion.html](http://www.appleinsider.com/articles/11/02/18/rim_nokia_and_googles_android_battle_for_apples_ios_scraps_as_app_market_sales_grow_to_2_2_billion.html).

Distimo.Huhtikuu 2010.April 2010 Publication.Viitattu 24.01.2011.
[Http://www.distimo.com/publications/](http://www.distimo.com/publications/).

Edis, S. 08.01.2011. Foorumikirjoitus mainostuloista. Viitattu 26.01.2011.
[Http://forum.unity3d.com/threads/73123-AdMob-statistics](http://forum.unity3d.com/threads/73123-AdMob-statistics).

Flurry. 14.10.2010. Madison Avenue and the Land of Make Believe. Viitattu 25.01.2011.
[Http://blog.flurry.com/bid/48418/Madison-Avenue-and-the-Land-of-Make-Believe](http://blog.flurry.com/bid/48418/Madison-Avenue-and-the-Land-of-Make-Believe).

Haselton, T. Mobileburn. 07.07.2010. Jobs discusses Apple App Store figures. Viitattu 24.01.2011.
[Http://www.mobileburn.com/news.jsp?id=9650](http://www.mobileburn.com/news.jsp?id=9650).

Lardinois, F. 11.08.2009. How do iPhone users find new apps? Viitattu 18.01.2011.

[Http://www.readwriteweb.com/archives/how_do_iphone_users_find_new_apps.php](http://www.readwriteweb.com/archives/how_do_iphone_users_find_new_apps.php).

Localytics. 31.01.2011. First Impressions Matter. Viitattu 26.01.2011.
[Http://www.localytics.com/blog/2011/first-impressions-matter-26-percent-of-apps-downloaded-used-just-once/](http://www.localytics.com/blog/2011/first-impressions-matter-26-percent-of-apps-downloaded-used-just-once/).

Murphy, M. Android Guys. 14.09.2009. 40 Android Business Models. Viitattu 23.01.2011.
[Http://www.androidguys.com/2009/09/14/40bizmodels/](http://www.androidguys.com/2009/09/14/40bizmodels/).

Northcott, G. 02.02.2010. Brand iPhone apps: benchmarks for success. Viitattu 24.01.2011.
[Http://geoffnorthcott.com/blog/2010/02/brand-iphone-apps-benchmarks-for-success/](http://geoffnorthcott.com/blog/2010/02/brand-iphone-apps-benchmarks-for-success/).

Parr, B. Mashable. 08.12.2010. Angry Birds hits 42 million free and paid downloads. Viitattu 22.01.2011.
[Http://mashable.com/2010/12/08/angry-birds-hits-42-million-downloads/](http://mashable.com/2010/12/08/angry-birds-hits-42-million-downloads/).

Perez, S. 17.01.2011. What Percentage of iPhone Owners are on iOS4? Viitattu 23.01.2011.
[Http://www.readwriteweb.com/mobile/2011/01/what-percentage-of-iphone-owners-are-on-ios4.php](http://www.readwriteweb.com/mobile/2011/01/what-percentage-of-iphone-owners-are-on-ios4.php).

Qt Developer Network. 2011. Qt:lle kehittäjien sivusto. Viitattu 12.1.2011.
[Http://developer.qt.nokia.com/](http://developer.qt.nokia.com/).

Van Zummeren, T. 25.10.2010. Developing apps compatible with all iOS devices. Viitattu 15.01.2011.
[Http://blog.jteam.nl/2010/11/25/developing-apps-compatible-with-all-ios-devices/](http://blog.jteam.nl/2010/11/25/developing-apps-compatible-with-all-ios-devices/).

Viestintävirasto. 27.09.2007. Digitaalinen allekirjoitus. Viitattu 18.01.2011.
[Http://www.ficora.fi/index/palvelut/palvelutaiheittain/tietoturva/pki/digitaalinen_allekirjoitus.html](http://www.ficora.fi/index/palvelut/palvelutaiheittain/tietoturva/pki/digitaalinen_allekirjoitus.html).

W3C. Maaliskuu 2011. HTML5 Määrittely. Viitattu 02.03.2011.
[Http://dev.w3.org/html5/spec/Overview.html](http://dev.w3.org/html5/spec/Overview.html).

W3i. 10.02.2011. How to top the App Store charts. Viitattu 28.02.2011.
[Http://blog.w3i.com/2011/02/17/how-to-top-the-app-store-charts-hint-get-your-ducks-in-a-row/](http://blog.w3i.com/2011/02/17/how-to-top-the-app-store-charts-hint-get-your-ducks-in-a-row/).

Wooldridge D., Schneider, M. 2010. The Business of iPhone App Development: Making and Marketing Apps that Succeed. Yhdysvallat: Apress.

Yarmosh, K. 2010. App Savvy: Turning Ideas into iPad& iPhone Apps Customers Really Want. 1 p. Yhdysvallat: O'Reilly.