



**HAAGA-HELIA**  
ammattikorkeakoulu

## **Windows Phone sovelluskehitys**

Anssi Metsäranta

Opinnäytetyö

23.10.2013



<b>Tekijä</b> Anssi Metsäranta	<b>Aloitusvuosi</b> 2009
<b>Opinnäytetyön nimi</b> Windows Phone sovelluskehitys	<b>Sivu- ja liitesivumäärä</b> 24
<b>Ohjaaja</b> Juhani Välimäki	
<p>Tämän opinnäytetyön aiheena on Windows Phone -sovelluskehitys. Työn teoriaosuus koostuu Windows Phone 8 -sovelluskehityksen perusteista ja valmiin sovelluksen julkaisusta sovelluskauppaan. Työ esittelee myös Windows Phone 8 -käyttöjärjestelmän ominaisuuksia ja käytettävissä olevia ohjelmointikieliä. Materiaalina teoriataustassa on käytetty aihetta käsittelevää kirjallisuutta, opinnäytetöitä ja MSDN:n verkkomateriaaleja.</p> <p>Tutkimusosiossa tarkastellaan kahdeksan eri opinnäytetyön kautta Windows Phonelle tapahtuvaa sovelluskehitystä.</p> <p>Älypuhelimien lisääntyvä määrä ja Windows Phone -käyttöjärjestelmän kasvava suosio antavat aihetta tutkia kyseiselle alustalle tapahtuvaa sovelluskehitystä.</p> <p>Tutkimustyön tuloksia voidaan käyttää Windows Phone -sovelluskehityksen aloitusvaiheessa. Työ antaa yleiskatsauksen Windows Phone -alustaan ja sen sovelluskehityksen vaatimuksiin.</p>	
<b>Asiasanat</b> Windows Phone, sovelluskehitys, sovelluskauppa, mobiilikehitys, Windows Phone 8	

<p><b>Author</b> Anssi Metsäranta</p>	<p><b>Year of entry</b> 2009</p>
<p><b>The title of thesis</b> Windows Phone application development</p>	<p><b>Number of pages and appendices</b> 24</p>
<p><b>Advisor</b> Juhani Välimäki</p> <p>The increasing numbers in smartphone sales and the rising popularity of the Windows Phone operating system gives a reason to study the application development for Windows Phone. The purpose of this thesis was to clarify the application development for the Windows Phone operating system.</p> <p>The theoretical part of the thesis consists of basic information about the Windows Phone software development and about the releasing of an application in the application store. The thesis also presents features of the Windows Phone 8 operating system and the programming languages available for development. The material used in the theoretical part includes Windows Phone development literature, theses and MSDN material.</p> <p>Eight different theses were examined in the practical part to give an overview about the Windows Phone application development.</p> <p>The results of this thesis can be used when beginning Windows Phone application development. The study provides an overview to the Windows Phone platform and its requirements for application development.</p>	
<p><b>Key words</b> Windows Phone, Mobile, Application development, Application store, Mobile development, Windows Phone 8</p>	

# Sisällys

Termit ja lyhenteet .....	3
Johdanto .....	4
1 Teoriatausta.....	5
1.1 Windows Phone -käyttöjärjestelmä.....	5
1.2 Windows Phone 8 Update 3 .....	6
1.3 Sovelluskehityksen järjestelmävaatimukset .....	6
1.4 Windows Phone ohjelmointikielet.....	7
1.4.1 C#.....	8
1.5 Windows Phone 8 SDK.....	8
1.6 Visual Studio .....	9
1.6.1 Visual Studio Express 2012.....	9
1.6.2 Blend .....	9
1.7 Emulaattori.....	10
1.8 XAML.....	10
1.9 Ohjelmointirajapinnat (API).....	10
1.10 Sovelluskauppa .....	11
1.11 Sovelluksen sertifiointi.....	11
1.12 Käyttöliittymä .....	13
1.13 Sovelluksen elinkaari (App lifecycle) .....	14
1.13.1 XNA ja MonoGame .....	16
2 Tutkimusmenetelmä .....	17
2.1 Tutkimuskysymykset.....	17
3 Tutkimus .....	18
4 Tulokset ja johtopäätökset.....	22
5 Yhteenveto .....	24
5.1 Jatkotutkimusehdotukset.....	24
Lähdeluettelo .....	25

## Termit ja lyhenteet

SDK	Software Development Kit, ohjelmiston kehittämiseen vaadittavat kehitystyökalut sisältävä paketti
Emulaattori	Kehitystyössä käytettävä työkalu, jonka avulla sovellusta voidaan testata ilman fyysistä laitetta
Sovelluskauppa	Window Phone Store
Blend	Käyttöliittymien suunnittelutyökalu
WP	Windows Phone
MSDN	Microsoft Developer Network, Microsoftin kehittäjäpalvelu
XAML	Extensible Application Markup Language, Sovelluksen käyttöliittymän kuvaava kieli
XNA	Peliohjelmointiin tarkoitettu ohjelmointikirjasto
API	Application programming interface, ohjelmointirajapinta sovellusten keskinäisille pyynnöille ja tiedonvaihdolle
Modern UI	Windows Phone 8:n käyttöliittymä

## Johdanto

Tämän opinnäytetyön aiheena on sovelluskehitys Windows Phone -käyttöjärjestelmälle. Työn tarkoitus on esitellä Windows Phone -käyttöjärjestelmää ja käytettyjen lähteiden pohjalta tutkia mitä vaaditaan sovelluksen kehittämiseen Windows Phone:lle. Tämä työ on rajattu käsittämään erityisesti Windows Phone 8:lle tapahtuvaa sovelluskehitystä. Aiempia käyttöjärjestelmäversioita ja niiden kanssa käytettäviä tekniikoita käsitellään teoriataustassa pintapuolisesti, mutta tämän työn osalta esitellyt sovelluskehityksen tekniikat ja vaatimukset keskittyvät Windows Phone 8:aan. Tutkimusosiossa tarkastellaan kahdeksan eri opinnäytetyön kautta vastaukset tutkimuskysymyksiin;

1. Kuinka sovellus kehitetään Windows Phone -käyttöjärjestelmälle?
2. Minkälaisia ovat Windows Phone -sovelluskehityksessä vaadittavat työkalut?
3. Kuinka valmis sovellus julkaistaan sovelluskauppaan?

Aihe on ajankohtainen, sillä Windows Phone on kirjoitushetkellä prosentuaalisesti suurimmassa kasvussa oleva mobiilikäyttöjärjestelmä.

Kiinnostus älypuhelimia kohtaan johti tämän aiheen valintaan. Kirjoittajalla itsellään ei ole aiempaa kokemusta tai tietoa Windows Phone sovelluskehityksen suhteen.

Materiaalina käytössä on ollut Windows Phone -sovelluskehitystä koskevaa kirjallisuutta ja Microsoftin kehittäjäpalvelu MSDN:n aineistoja.

# 1 Teoriatausta

Tässä luvussa esitellään lyhyesti Windows Phone -käyttöjärjestelmä, esitellään Windows Phone -sovelluskehityksen perusedellytykset ja kuinka valmis sovellus julkaistaan sovel-luskaupassa.

## 1.1 Windows Phone -käyttöjärjestelmä

Windows Phone on Microsoftin luoma mobiilikäyttöjärjestelmä, jonka se julkisti vuonna 2010. Windows Phone kehitettiin pitkälti tyhjältä pöydältä. Yhteistä edeltäjänsä (Win-dows Mobile), sillä on käyttöjärjestelmän ytimessä käytettävä Windows CE:stä sovitettu koodi. Merkittävänä erona taas, etteivät Windows Mobilelle kehitetyt sovellukset toimi Windows Phonessa suoraan (Järvinen) (Lindholm).

Windows Phone 8 on käyttöjärjestelmän viimeisin versio. Windows Phone 7:ssä käytetty Windows CE- pohjainen arkkitehtuuri on tässä korvattu Windows NT- pohjaisella käyt-töjärjestelmäytimellä. Samaa käyttöjärjestelmäydintä käyttää myös Windows 8- käyttöjär-jestelmä. Windows Phone 7 – käyttöjärjestelmän omaavia laitteita ei voida päivittää WP 8:aan (Pahkala) (Kuusinen).

Käyttöjärjestelmäytimen vaihto toi mukanaan muutoksia käyttöjärjestelmän arkkitehtuu-riin, laitteistovaatimuksiin ja ominaisuuksiin. Yhteneväisyys Windows 8 käyttöjärjestel-mäytimen kanssa tarkoittaa, että sovellusta kehittäessä voidaan samaa koodia käyttää osittain kummallekin käyttöjärjestelmälle. Vaihdon myötä saatiin myös tuki mo-niydinprosessoreille ja uusille resoluutioille (1280 x 720 ja 1280 x 768). (Pahkala) (Kuusinen). Sovelluskehittäjän näkökulmasta Windows Phone 8:n mukana on tullut myös tuki C++ -ohjelmoinnille.

Tutkimusyhtiö Gartnerin mukaan Windows Phone on prosentuaalisesti mitaten vuoden 2013 kolmannella neljänneksellä suurimmassa kasvussa oleva mobiilikäyttöjärjestelmä. Windows Phone on Androidin ja iOS:in jälkeen kolmanneksi suurin käyttöjärjestelmä 3,6 prosentin osuudellaan. Käyttöjärjestelmää ei ole rajoitettu määrättyyn valmistajaan,

vaan tällä hetkellä Windows Phone 8 -käyttöjärjestelmällä varustettuja puhelimia valmistavat Huawei, Nokia, Samsung ja HTC. (Gartner)

## **1.2 Windows Phone 8 Update 3**

Tätä opinnäytetyötä kirjoitettaessa käyttöjärjestelmän viimeisin versio on 8.0.10512 (Windows Phone Update 3), joka toi mukanaan muun muassa 1080p- resoluution ja tuen uudelle Qualcommin Snapdragon 800- neliydinsuorittimelle. Päivitys on julkaistu 14.10.2013 ja se on aluksi saatavilla vain sovelluskehittäjille uuden Developer Preview ohjelman kautta. Näin kehittäjät pääsevät kokeilemaan uusia ominaisuuksia, ennen kuin ne tulevat tavallisten käyttäjien puhelimiin. Kehittäjien Preview – päivitys ei tosin vielä sisällä aivan niitä kaikkia ominaisuuksia, jotka löytyvät aikanaan valmiista päivityksestä. Tulevien kuukausien aikana päivitys saapuu myös kuluttajien saataville (Windows Phone Developer Blog)

## **1.3 Sovelluskehityksen järjestelmävaatimukset**

Windows Phone 8 -sovelluskehitys vaatii toimiakseen Windows Phone 8 SDK:n. Käyttöjärjestelmän tulee olla 64-bittinen Windows 8 ja tästä tarkennuksena mainittakoon, että Windows Phone emulaattori vaatii toimiakseen Windows 8 Pro -version. Tallennustilaa vaaditaan 6,5 GB ja keskusmuistia 4GB. (Kuusinen) (MSDN)

Windows Phone sovelluskehityksen tärkein kehitysväline on Visual Studio. Windows Phone 8 SDK sisältää erillisen Visual Studio Express for Windows Phone -nimisen version kehittäjästä. Mikäli kehittäjältä kuitenkin on jo käytössään Visual Studio 2012 Pro, Premium tai Ultimate – versio, niin tulee siihen WP -kehitystä varten ladata puhelintyökalut - Phone SDK (Kuusinen).



Ohessa Microsoftin edellyttämiä Windows Phone 8 -laitteiden minimivaatimuksia:

Qualcomm Snapdragon S4 kaksisydinprosessori tai Snapdragon 800 prosessori
Keskusmuistia vähintään 512MB 800×480 – resoluutioisille laitteille – isommat resoluutiot vaativat 1GB keskusmuistia
Flash – muistia 4GB
Tuki micro-USB 2.0 liitännälle
GPS-vastaanotin, kiihtyvyyssanturi, valaistuksen- ja etäisyydentunnistus

Kehittäjä voi käyttää joko Express tai Professional-versioita kehittimistä. Express – versiot kehittimistä ovat maksuttomia, mutta ne eivät tarjoa kaikkia samoja ominaisuuksia kuin maksulliset versiot. Windows Phone sovelluskehittäjän tuleekin tietää, mitä ominaisuuksia sovellukselta vaaditaan. Visual Studio Express versiolla on mahdollista kehittää ainoastaan puhelimesta toimivia sovelluksia – esimerkiksi webpohjaiset taustajärjestelmät eivät ole sillä mahdollisia toteuttaa (Lindholm).

#### 1.4 Windows Phone ohjelmointikielät

Windows Phone 8 sovelluksia voidaan toteuttaa muutamalla eri ohjelmointi ja merkintäkielen yhdistelmällä. Vaihtoehtoina on: C# tai Visual Basic ja XAML sekä C++ ja DirectX/XAML -yhdistelmät. Lisäksi JavaScript ja HTML 5 -yhdistelmä on mahdollinen, joskin JavaScriptiä ei varsinaisesti tueta Windows Phone 8:ssa (Kuusinen). XAML:lla toteutetaan sovelluksen ulkoasu ja varsinainen ohjelmointi jollain edellä mainituista kielistä. Esimerkiksi peliohjelmoinnissa C++ ja DirectX -yhdistelmä on suosittu (Kuusinen).

Microsoftin julkaisemasta taulukosta löytyvät Windows Phonen ja Windows 8:n tukemat kielet (Kuva 1). C# ja Visual Basicin (VB.NET) kanssa merkintäkielenä käytetään XAML:ää. C++ sovelluksessa voidaan käyttää joko XAML:ää tai DirectX:ää. JavaScript ja HTML 5 -yhdistelmä on myös mahdollinen esimerkiksi PhoneGap -nimisen kehityskirjaston avulla.

Windows Phone -ohjelmistokehityksessä suosituin kieli on C# (Järvinen).

	.NET (C#, VB.NET)	C++	JAVASCRIPT
WINDOWS PHONE 8	✓	✓	
WINDOWS 8	✓	✓	✓

Kuva 1 Windows Phonen ja Windows 8:n tukemat kielet

### 1.4.1 C#

C# on Microsoftin kehittämä oliopohjainen ohjelmointikieli ja se on osa .NET kehitysalustaa. Kieli sisältää monia C, C++, Visual Basic ja Java -kielten ominaisuuksia (Klätik)

## 1.5 Windows Phone 8 SDK

Windows Phone 8 SDK (Software Developer Kit) on ohjelmapaketti sovelluksen kehitystyöhön. Kyseinen paketti sisältää seuraavat työkalut:

- Visual Studio Express 2012 for Windows Phone
- Blend for Visual Studio 2012
- Windows Phone -emulaattori

(The Visual Studio Blog)

## 1.6 Visual Studio

Visual Studio on Microsoftin pääsovelluskehitysväline. Visual Studio sisältää työkalut sovelluskehitykseen ja käyttöliittymäsuunnitteluun. Sovelluskehitys tapahtuu Visual Studiolla ja käyttöliittymäsuunnitteluaan Blendillä. Visual Studio sisältää kyllä perustyökalut käyttöliittymien toteuttamiseen, mutta mikäli halutaan graafisesti näyttävämpiä sovelluksia, tulisi käyttää lisäksi Blendiä (Kuusinen). Visual Studio on rad-periaatteeseen perustuva kehitin (Rapid Application Development). Sen perusosat ovat koodieditori, visuaalinen käyttöliittymien suunnittelu sekä virheenjäljitystoiminnot (Järvinen).

### 1.6.1 Visual Studio Express 2012

Microsoftin pääsovelluskehitysväline on Visual Studio. Tämä WP 8 SDK:n mukana tuleva versio kyseisestä kehittäjätyökalusta on nimeltään Visual Studio Express 2012 for Windows Phone. Kyseinen kehitin sisältää nimensä mukaisesti vain mobiilisovelluksen kehitykseen vaadittavat ominaisuudet. (Järvinen) (Pahkala)

### 1.6.2 Blend

Blend (aiemmin Expression Blend) on sovellusten graafisten käyttöliittymien ja animaatioiden luomiseen tarkoitettu työkalu. Blendin tarkoituksena on, että sovelluksen käyttöliittymän suunnittelija voi toteuttaa haluamansa näköisen käyttöliittymän ja samanaikaisesti sovelluskehittäjä luo sovelluksen varsinaisen ohjelmakoodin Visual Studiolla. (Pahkala)

## 1.7 Emulaattori

Windows Phone Emulator on emulaattori – työkalu, joka sisältyy WP 8 SDK:hon. . Tämä työkalu emuloi WP-laitetta, joten sen avulla voidaan WP- sovellusta testata virtuaaliympäristössä, ilman fyysistä WP-laitetta. Emulaattori pitää sisällään käyttöjärjestelmän oleelliset toiminnot, muttei esimerkiksi puheluiden soittoa tai tekstiviestien lähetystä. Emulaattorin toiminnot ovat kuitenkin niin samankaltaisia kuin fyysisen laitteenkin, että sovellusten testaus ja kehitys on sen avulla mahdollista. Emulaattorin toiminta-vaatimuksena on Windows 8:n Pro versio. Syy siihen on, että emulaattori käyttää Pro-version sisältämää Hyper-V- virtualisointialustaa (Kuusinen) (Järvinen) (MSDN).

## 1.8 XAML

XAML (Extensible Application Markup Language) on Microsoftin kehittämä XML-pohjainen merkintäkieli. Kyseinen kieli on deklaratiiivinen, eli se kuvaa mitä toiminnot tarvitsevat toimiakseen, eikä sitä, kuinka ne tulisi toteuttaa. WP sovelluskehityksessä XAML on oleellisessa osassa – sillä kuvataan sovelluksen käyttöliittymä ja siihen liittyvä toiminnallisuus (Kuusinen) (Järvinen). Visual Studiolla ja Blendillä XAML -koodin kirjoitus on mahdollista kahdella tavalla:

1. Kehittimien työkalupaletista vedetään halutut kontrollit suoraan sovelluksen käyttöliittymään, jolloin kehittimet luovat XAML -koodin.
2. Kirjoitetaan XAML -koodi käsin, jolloin kehittimet päivittävät sovelluksen käyttöliittymän vastaamaan tuota kirjoitettua koodia.

## 1.9 Ohjelmointirajapinnat (API)

Ohjelmointirajapinta (API, Application programming interface) luo sovelluskehittäjälle mahdollisuuden käyttää jonkin ohjelman yleisimpiä toimintoja. Toisin sanoen API toimii välikätenä sovelluskehittäjän sovelluksen ja jonkin valmiin ohjelman väliseen kommunikointiin (Lindholm).

## 1.10 Sovelluskauppa

Sovellusten asennus WP – laitteeseen onnistuu ainoastaan Windows Phone Store – nimisen sovelluskaupan kautta. Store sisältää sekä ilmaisia että maksullisia sovelluksia. Kehitystyökalujen (SDK) lataaminen ja käyttäminen on ilmaista, mutta mikäli sovellus halutaan saataville sovelluskauppaan ja sitä kautta laitteisiin asti, tulee sovelluskehittäjän hankkia tili Windows Phone Dev Center – palveluun. Kyseinen kehittäjätili maksaa vuodessa 14 euroa yksityishenkilöltä ja 75 euroa yritykseltä. Tilin luominen vaatii luottokortin, jonka avulla todennetaan käyttäjän henkilöllisyys. Voimassa olevat hinnat ja edellytysvaatimukset voi tarkistaa osoitteesta: <http://msdn.microsoft.com/en-us/library/windows/apps/jj863494.aspx>.

Kehittäjä voi itse määrittellä onko sovellus ilmainen vai maksullinen. Mikäli sovellus on maksullinen, ottaa Microsoft myyntituotosta 30 %.









## 1.11 Sovelluksen sertifiointi

Sovelluksen julkaisu Storeen vaatii Microsoftin sertifiointiprosessin läpikäymistä. Sertifiointinnissa tarkistetaan, että sovellus vastaa asettuja rajoituksia tekniikan ja sisältönsä suhteen. Tämä prosessi varmistaa, ettei sovellus sisällä esimerkiksi haittaohjelmia. Sertifiointivaatimukset löytyvät MSDN -kehittäjäpalvelusta osoitteesta:

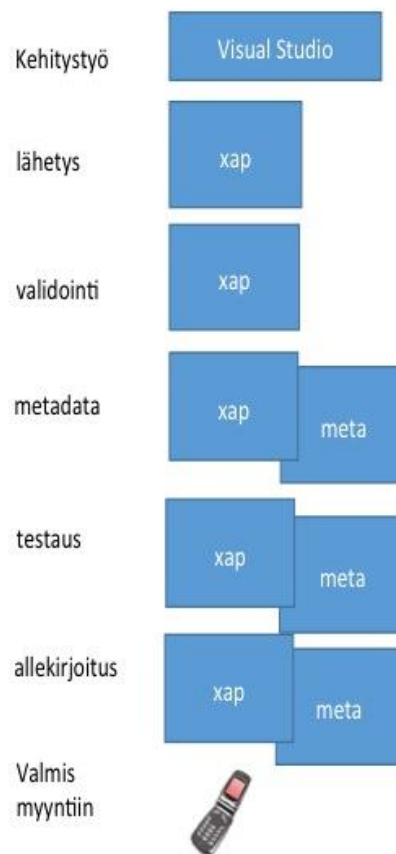
[http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh184843(v=vs.105).aspx) (Järvinen) (Kuusinen)

Valmis sovellus lähetetään Visual Studiolla xap- tiedostona sovelluskauppaan. Xap-tiedosto muodostuu automaattisesti, sovelluksen käännösvaiheessa (build). Tiedoston lisäksi tulee lähettää vaaditut kuvamateriaalit. Sovelluksen lähetys tapahtuu kahdeksassa vaiheessa (Kuva 2). Näissä vaiheissa muun muassa nimetään sovellus, määritellään sovelluskuvaus, sovelluksen hinta ja myyntimaat. Lopuksi sovellus lähetetään eteenpäin tarkistusta varten. Microsoftin sertifiointi kestää muutamasta päivästä viikkoihin. Mikäli sovellus ei läpäise sertifiointia, saa kehittäjä Microsoftilta raportin korjausta vaativista seikoista. Mahdollisen raportin korjaukset suoritettuaan voi kehittäjä lähettää sovelluk-

sen uudelleen arvioitavaksi. Sertifiointin läpäissyt sovellus ilmestyy sovelluskauppaanlatautavaksi (Lindholm) (Kuusinen)

 2 minuuttia	<b>Sovelluksen nimi</b> Anna sovellukselle yksilöllinen nimi. <a href="#">Lisätietoja</a>
 5 minuuttia	<b>Myyntitiedot</b> Valitse sovelluksen hinta, luettelon luokat sekä kohteet, joissa haluat myydä sitä. <a href="#">Lisätietoja</a>
 5 minuuttia	<b>Lisäominaisuudet</b> Määritä pikailmoitukset, Live-palvelut ja sovelluksen kautta tehtävät tarjoukset. <a href="#">Lisätietoja</a>
 5 minuuttia	<b>Ikäluokitus ja luokitusvarmenteet</b> Kuvaile sovelluksen kohderyhmä ja lataa luokitusvarmenteet. <a href="#">Lisätietoja</a>
 5 minuuttia	<b>Salaus</b> Määritä, tukeeko sovellus salausta, ja ota käyttöön paketin lataus. <a href="#">Lisätietoja</a>
 30 minuuttia	<b>Paketit</b> Lataa sovellus Windows-kauppaan. Ota tämä vaihe käyttöön täyttämällä salaussivu. <a href="#">Lisätietoja</a>
 30 minuuttia	<b>Kuvaus</b> Kuvaile lyhyesti sovellustasi asiakkaille. <a href="#">Lisätietoja</a>
 2 minuuttia	<b>Huomautukset testaajille</b> Lisää huomautuksia tästä julkaisusta sovelluksen tarkastajille. <a href="#">Lisätietoja</a>

Kuva 2 Sovelluksen lisääminen sovelluskauppaan (Kuusinen)



Kuva 3 Sovelluksen levityksen vaiheet (järvinen)

## 1.12 Käyttöliittymä

Windows Phone – käyttöjärjestelmässä sovellusten näyttävät ulkoasut ovat tärkeitä (järvinen). Sovellusten käyttöliittymät suunnitellaan Blend – työkalulla. Windows Phonen käyttöliittymästä käytetään nimitystä Modern tai Modern UI (aiemmin nimellä Metro).

Modern UI koostuu sovellusikoneista eli erikokoisista suorakaiteen muotoisista laati-koista, joita kutsutaan joko tiiliksi (tiles) tai tapahtumaruuduiksi (Kuva 4). Sovellus käynnistetään tiiltä koskettamalla. (Järvinen) (Kuusinen)

Tiilet itsessään ovat aktiivisia. Esimerkiksi sähköpostisovellus voi kertoa saapuneiden viestien määrän tiilessä.

Microsoft on ohjeistanut tarkkaan, millaisilta Modern UI:n mukaiset sovellukset tulisivat näyttää. Yhteneväisen käyttöliittymätyylin tarkoituksena on, että käyttäjän on helpompi omaksua sovellusten käyttö. Rungas dokumentaatio käyttöliittymäsuunnittelusta löytyy MSDN- kehittäjäpalvelusta : [MSDN UI Guidelines](#) (Järvinen)

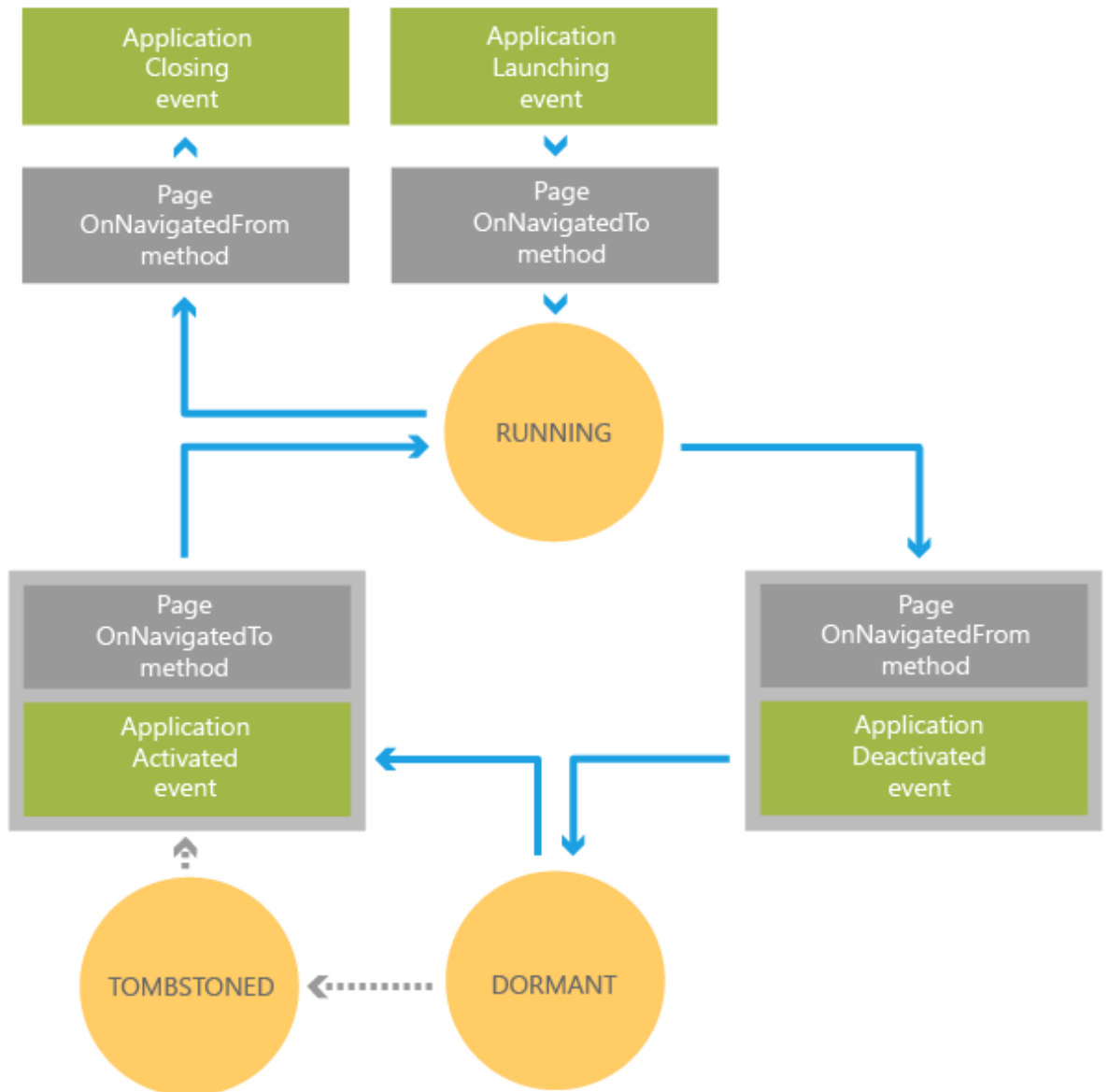


Kuva 4 Windows Phonen käyttöliittymä: Modern UI

### 1.13 Sovelluksen elinkaari (App lifecycle)

Sovelluksilla on tarkasti määritellyt elinkaaret, joita ne noudattavat. Sovelluksen elinkaari koostuu tyypillisesti käynnistyksestä, käyttö (running) ja sulkemisesta. Käyttöjärjestelmä kutsuu automaattisesti kunkin tilan mukaista koodia, sen perusteella mitä sovelluksessa tapahtuu (Duffy)





Kuva 5 Sovelluksen elinkaari (kuva msdn)

Windows Phone 8 sovelluksen tilat ovat Running, Dormant ja Tombstoned. Sovelluksen tilat on esitetty ohessa (Kuva 5).

Windows Phone sallii kerrallaan yhden sovelluksen käytön etualalla (foreground). Kun käyttäjä siirtyy pois käyttämästään sovelluksesta, niin sovellus menee Dormant -tilaan. Uinuvassa tilassa (dormant), sovellus ei ole enää varsinaisesti käynnissä, vaan sen säikeet ovat pysäytettyinä ja tiedot säilyvät muistissa. Kun käyttäjä palaa sovellukseen käyttöjärjestelmä hakee tiedot muistista ja sovellus palaa käynnissä- tilaan (running). Haudatuss-tilassa (Tombstoned), sovellus ei enää ole käynnissä, mutta järjestelmä säilyttää tietoja

sen aiemmasta tilasta. Haudattu- tilassa voi olla enintään viisi sovellusta kerrallaan. Kun käyttäjä palaa haudattuun sovellukseen, hakee käyttöjärjestelmä tiedot muistista ja käynnistää sovelluksen uudelleen (MSDN).

### **1.13.1 XNA ja MonoGame**

XNA on Microsoftin kehittämä peliohjelmointiin keskittyvä tekniikka. Kyseisen tekniikan avulla voidaan tehdä pelejä myös Xbox- ja pc- alustoille. XNA:ta ei voida käyttää WP 8 -sovelluskehityksessä ja Microsoft on luopunut kyseisen tekniikan kehityksestä. Mikäli XNA- kirjastoa halutaan käyttää, voidaan kehitys toteuttaa MonoGamella. MonoGame on avoimen lähdekoodin toteutus, joka pyrkii tuomaan XNA-kirjaston Windowsin lisäksi myös muille alustoille (Kuusinen).

## 2 Tutkimusmenetelmä

Tämän tutkimuksen menetelmänä käytetään kirjallisuustutkimusta. Tutkimus suoritetaan osana tutkimusryhmää, jossa määritellyn aikataulun vuoksi myös työn laajuus on rajoitettu.

Kirjallisuustutkimus on tutkimusprojektin kannalta tärkeä ensiaskel. Sen avulla kartoitetaan haluttu tieto ja samalla selvitetään, ettei asiaa ole jo sellaisenaan tutkittu. Kirjallisuustutkimuksen tavoitteena on vastata esitettyihin tutkimuskysymyksiin hankitun tutkimusaineiston analysoinnilla (vom Brocke).

### 2.1 Tutkimuskysymykset

Työ selvittää aiheeseen liittyvän kirjallisuuden ja lähdemateriaalin kautta vastauksen seuraaviin kysymyksiin:

1. Kuinka sovellus kehitetään Windows Phone -käyttöjärjestelmälle?
2. Minkälaisia ovat Windows Phone -sovelluskehityksessä vaadittavat työkalut?
3. Kuinka valmis sovellus julkaistaan sovelluskauppaan?

### 3 Tutkimus

Tutkimuksessa tarkasteltiin lähemmin kahdeksaa eri Windows Phone -sovelluskehitystä käsittelevää opinnäytetyötä. Näissä jokaisessa opinnäytetyössä on kehitetty toimiva WP -sovellus. Sovellukset ovat keskenään erityyppisiä ja hyödyntävät alustan eri ominaisuuksia.

#### Windows Phone 8 sovellukset

1. Kuusinen: ristinolla-pelisovellus (Kuusinen)
2. Pahkala: esimerkki kameraa ja sijaintipalvelua käyttävästä sovelluksesta (Pahkala)
3. Becks: RSS-syötteiden lukija (Becks)
4. Voroshilova: Sijaintipalvelusovellus (Voroshilova)

#### Windows Phone 7 ja 8 -sovellukset

5. Timonen: Ampparit.com -sovellus (Timonen)

#### Windows Phone 7.5 (kehitetty WP 8 -SDK:lla)

6. Klátik: Greenriders -sovellus (Klátik)

#### Windows Phone 7 sovellukset:

7. Lindholm: Twitter -sovellus (Lindholm)
8. Peltoniemi: Reittiopas-sovellus (Peltoniemi)

Tutkimuksessa ensimmäinen tutkittava sovellus on Niko Kuusinen opinnäytetyössä kehitetty Ristinolla-pelisovellus. Kyseessä on verkon yli pelattava kaksinpeli, joka sisältää keskustelumahdollisuuden toisten pelaajien kanssa. Kehitetty peli toimii sekä Windows Phone 8:ssä että Windows 8:ssä. Peli on toteutettu C# ja XAML -kielten yhdistelmällä. Kielet on tässä tapauksessa valittu kehittäjien aiempien tietotaidon perusteella. Pelin verkkototeutus on rakennettu Windows Communication Foundation -tekniikalla ja pelin käyttämät palvelut on sijoitettu Microsoftin Azure -pilvipalveluun. XNA -pelikirjastojen käytöstä luovuttiin kun oli havaittu, ettei peli tarvitse niitä. Sovelluskehityksessä käytössä oli kolme kappaletta WP 8 -puhelimia. Näiden puhelimien kanssa ilmeni testivaiheessa joitain pieniä ongelmia sovelluksen verkkototeutuksen suhteen. Emulaattorin toiminnassa näitä ongelmia ei kuitenkaan havaittu (Kuusinen).

Jan Pahkala esittelee työssään esimerkit kameraa käyttävästä sovelluksesta ja sijaintipalvelua käyttävästä sovelluksesta. Esimerkit on toteutettu C# ja XAML -kielillä käyttäen Visual Studio 2012 Express for Windows Phone 8 -kehittäjä. Kumpikin sovellus on testattu toimivaksi emulaattorissa, eikä niitä ole julkaistu sovelluskaupassa (**Pahkala**).

Olli Timosen työ on Ampparit.com-sovellus, joka näyttää jo olemassa olevan Ampparit.com HTML5-sivuston omassa Windows Phone -sovelluksessaan. Sovellus on kehitetty Visual Studiolla käyttämällä XAML- ja C#-ohjelmointikieliä. Sovellus on julkaistu kauppapaikassa. Sovelluksen luokkarakenteeseen mallia on otettu sen valmiista Android-versiosta. Android -version lähdekoodin kääntämisestä oli kehitysvaiheessa luovuttu ilmenneiden ongelmien vuoksi ja koko sovelluksen rakenne toteutettiin uudelleen. HTML5 -tuen erot käyttöjärjestelmäversioiden välillä aiheuttivat yhteensopivuusongelmia, joten lopulta kehitettiin oma sovellus sekä Windows Phone 7:lle että 8:lle (**Timonen**).

Tuomas Becksin sovellus on RSS-syötteiden lukijan prototyyppi. Sovellus on toteutettu C# ja XAML-kielillä. Sovellus käyttää SQLite:tä tietokantarakaisunaan. Kehittäjällä ei ollut aiempaa kokemusta kyseisistä kielistä tai Windows Phone -sovelluskehityksestä, mutta aiempi Javan tuntemus oli kehittäjän mielestä hyödyksi. Sovellusta ei ole toistaiseksi julkaistu sovelluskaupassa, mutta sitä on tarkoitus jatkossa kehittää julkaisukelpoiseksi (**Becks**).

Alexandra Voroshilovan sovellus on sijaintipalvelusovellus. Sovelluksen ajatuksena on lähettää automaattisesti ilmoitusviestejä käyttäjän haluamille vastaanottajille, kun saavutaan määrättyyn sijaintiin. Sovellusta voidaan esimerkiksi vanhemmille, jotka haluavat tietää kun lapsi on saapunut kotiin. Sovelluksen tiedonvaihtoon käyttämät toiminnot on sijoitettu Azure -pilvipalveluun. Sovellusta ei ole julkaistu sovelluskaupassa (**Voroshilova**).

Klátik Martinin kehittämä sovellus on nimeltään Greenriders. Sovellus on palvelu, jonka avulla ihmiset voivat löytää ja luoda kimppakyytejä esimerkiksi työmatkoilleen. Sovellus on opinnäytetyössä kehitetty Windows Phone 7.5:lle, mutta se on tarkoitus kehittää myöhemmin Windows Phone 8:lle. Sovellus on toteutettu C# ja XAML -kielillä käyttäen Visual Studio 2012 Express for Windows Phone 8 -kehittäjä. Sovellusta ei ole julkaistu sovelluskaupassa, sillä sen kehitys jatkuu edelleen (Klátik)

Joonas Lindholmin opinnäytetyössä on kehitetty esimerkki Twitteriä käyttävästä sovelluksesta Windows Phone 7:lle. Sovellusta ei ole julkaistu kauppapaikassa, sillä kehittäjä ei halunnut maksaa tuolloin käytössä ollutta 99 dollarin kehittäjämaksua. Sovellus on toteutettu C# ja XAML -kielillä. Sovellus käyttää Twitterin REST API:a Twitterin toimintojen käyttöä varten (Lindholm).

Erkka Peltoniemen Reitti 1.0 - on reittiopassovellus Helsingin Seudun Liikenteen (HSL) palveluihin. Sovellus käyttää Reittiopas API:a HSL:n aikataulujen hankkimiseen. Puhelimen sijaintipalveluita hyödynnetään käyttäjän sijainnin paikantamisessa, jolloin käyttäjä voi hakea reittejä julkisen liikenteen alueella olevaan sijaintiin. Sovellus hyödyntää myös puhelimen kameraominaisuutta. Käyttöliittymäsuunnittelussa on pyritty noudattamaan mahdollisimman hyvin Windows Phonen suosituksia. Sovellus on toteutettu C# ja XAML -kielillä. Sovellus on julkaistu kauppapaikassa (Peltoniemi).

Sovellus	Laitteisto-ominaisuudet (Kamera, kompassi, GPS)	Rajapinnat muihin palveluihin	Julkaistu sovel-luskaupassa
Ristinolla-sovellus		x	x
Sijaintipalveluesimerkki	x		
Kameraesimerkki	x		
Ampparit.com -sovellus			x
Sijainpalvelusovellus	x	x	
RSS-lukija		x	
Greenriders -sovellus	x	x	
Twitter-sovellus		x	
Reittiopas-sovellus	x	x	

Taulukko 1 Tutkitut sovellukset

Taulukkoon (**Error! Reference source not found.**) on listattu mitkä tutkituista sovelluksista hyödynsivät WP -puhelimien laitteisto-ominaisuuksia, mitkä sovellukset käyttivät rajapintoja muihin palveluihin ja oliko niitä julkaistu sovelluskaupassa.

## 4 Tulokset ja johtopäätökset

Windows Phone -sovelluskehityksessä suosituin kieli on C#. Tämä käy ilmi niin teoriataustassa kuin tutkituissa sovelluksissa. Windows Phone 8 -sovelluskehitykseen tarjolla olevasta materiaalista suurin osa on C# ja XAML:ään keskittyvää, joten varsinkin aloittelevalle sovelluskehittäjälle se on varmasti jatkossakin hyvä vaihtoehto. Ala on kuitenkin jatkuvasti elävä, joten sovelluskehittäjän onkin hyvä osata ainakin perusteet useammasta kielestä. C# -ohjelmoinnin aloittaminen on helpompaa, mikäli aiempaa kokemusta löytyy esimerkiksi Javasta tai C++:sta. Oleellista on toki se, minkälaista sovellusta halutaan kehittää. Mikäli kyseessä on näytävä pelisovellus, niin C++ voi olla järkevä ratkaisu.

MSDN -kehittäjäpalvelun materiaalit ja se sisältää paljon selkeitä käytännön esimerkkejä. Suurin osa MSDN:n esimerkeistä on C# ja XAML -pohjaisia. Windows Phone -sovelluskehityksen liittyvä kirjoitettu aineisto ja tarjolla olevat verkkomateriaalit ovat vielä monelta osin Windows Phone 7 -sovelluskehitystä käsitteleviä, mutta tämä selittyy luonnollisesti WP 8 -alustan tuoreudesta. Yhteneväisyyksiä näiden järjestelmäversioiden välillä on toki sen verran paljon, että vanhemmastakin materiaalista on sovelluskehityksessä hyötyä.

Visual Studio on Windows Phone sovelluskehityksessä käytettävä kehitin. Sovelluskehityksessä tehtävän vaatimusmäärittelyn pohjalta voidaan pohtia riittäkö oman sovelluksen kehittämiseen maksuttomat työkalut vai vaatiiko jokin sovelluksen toiminto maksullisia kehittäjiä. Tutkimuksen sovelluskehittäjät mainitsevat kehitystyökalut helppokäyttöisiksi ja vakaiksi.

Microsoftin Windows Phone -sovelluskauppa on Windows Phone -sovellusten ainoa jakopiste, eikä ainakaan toistaiseksi mikään viittaa, että tähän asiaan olisi tulossa muutoksia. Tutkittujen sovellusten kehittäjät olivat kommentoissaan varsin yksimielisiä siitä, että sovelluskauppaan sovelluksen lisääminen on kuitenkin yksinkertaista.

Tutkimuksesta käy ilmi, että Windows Phone 8 on tuonut edeltäjänsä verrattuna paljon uusia ominaisuuksia sovelluskehittäjälle. Esimerkiksi HTML 5 tuki on WP 8:ssa selvästi suurempi kuin WP 7:ssä, mutta vaatii edelleen parantamista.



Käyttöjärjestelmän kasvava suosio voi olla sovelluskehittäjän näkökulmasta kiinnostava asia.

Kirjallisuustutkimus on hyvä aloitus uuden asian tutkimisprosessissa. Tämän pohjalta saatiin selvitettyä Windows Phone -sovelluskehityksen perusteet. Tutkijalla itsellään ei ollut aiempaa käsitystä Windows Phone -sovelluskehityksestä. Myös tämä seikka puolsi kirjallisuustutkimuksen tekemistä aiheesta. Työ kokonaisuudessaan antaa selkeän yleiskuvan Windows Phone -käyttöjärjestelmästä ja sille tapahtuvasta sovelluskehityksestä. Luonnollinen jatkumo tutkimukselle olisi kartoitetun tiedon pohjalta oman sovelluksen kehittäminen ja julkaisu.

Työ toteutettiin osana tutkimusryhmää, jolloin aikataulullisten resurssien niukkuus määritteli pitkälti tutkimuksen laajuutta. Mikäli aikaa olisi ollut enemmän, olisi kirjallisuustutkimuksessa tutkittujen sovellusten määrä voinut olla suurempi. Työstä jouduttiin rajamaan pois monia hyödyllisiä asioita, esimerkiksi Windows Phone -laitteisto-ominaisuuksista olisi voinut olla enemmän tietoa. Toisaalta tiukat aikataulut mahdollistivat, että työ todella tuli tehtyä ajallaan valmiiksi.

## 5 Yhteenveto

Työn tavoitteena oli tutkia Windows Phone -käyttöjärjestelmälle tapahtuvaa sovelluskehitystä. Työlle asetetut tiukat aikataulut rajoittivat tutkimuksen laajuutta. Aihe on itsessään hyvin laaja, joten syvällisempikin katsaus olisi varmasti ollut mahdollinen. Tutkijalla itsellään ei ollut aiempaa tietoa Windows Phone -sovelluskehityksestä, joten kirjallisuustutkimuksen toteuttaminen tiedon keräämiseksi oli hyvä ratkaisu.

Mobiiliala on jatkuvan muutoksen kourissa ja historia onkin osoittanut, etteivät alansuurienkaan toimijoiden ja niiden järjestelmien asema ole kiveen hakattu. Windows Phone on kuitenkin tällä hetkellä kasvussa oleva alusta, joten oli hyvä hetki kartoittaa sen sovelluskehityksen perusteita.

Mikäli aikaa olisi ollut enemmän, olisi ollut mielenkiintoista testata itse käytännössä sovelluksen kehitystä ja julkaisua. Windows Phone on alustana kasvavassa suosiossa, joskin Googlen Androidin ja Applen iOS:in etumatka on vielä melkoinen.

Teoriatausta ja tutkimus osoittavat, että ohjelmointikielinä C# ja XAML -yhdistelmä on ainakin toistaiseksi suosituin vaihtoehto Windows Phone -sovelluskehityksessä. HTML 5 -tuki tulee varmasti jatkossakin lisääntymään mobiilikäyttöjärjestelmien suhteen ja sitä kautta myös tuki Windows Phone -alustalla tulee parantumaan.

### 5.1 Jatkotutkimusehdotukset

Tälle opinnäytetyölle asetettujen aikataulurajojen puitteissa jätettiin Windows Phone -sovelluskehityksen kokeileminen tutkijan osalta pois. Tutkimuksen pohjalta luonnollinen jatkotutkimuskohde olisikin oman sovelluksen kehittäminen ja julkaisu. Windows Phone 8 ja Windows 8 sovelluskehityksen eroista ja yhteneväisyyksistä saisi myös varmasti tutkimuksen aikaan.

## Lähdeluettelo

Windows Phone Developer Blog. (2013). *Introducing Windows Phone Preview for Developers*. Retrieved 11 26, 2013, from [http://blogs.windows.com/windows\\_phone/b/wpdev/archive/2013/10/14/introducing-windows-phone-preview-for-developers.aspx](http://blogs.windows.com/windows_phone/b/wpdev/archive/2013/10/14/introducing-windows-phone-preview-for-developers.aspx)

vom Brocke, S. N. (2009). *Reconstucting the Giant: on the Importance of Rigour in Documenting the Literature Search Process*.

Voroshilova, A. (2013). *Developing a location based application*.

Becks. (2013). *RSS-syötteiden lukijan totutus Windows Phone 8 -alustalle* .

Duffy. (2013). *Programming with mobile applications*. Boston: Course Techonology.

Gartner. (2013). *Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013*. Retrieved 11 26, 2013, from <http://www.gartner.com/newsroom/id/2623415>

Järvinen. (2012). *Windows Phone sovelluskehitys*. Porvoo: Bookwell.

Kuusinen. (2013). *Windows 8 Store- ja Windows Phone 8 –sovelluskehitys C#-ohjelmointikielellä ja XAML-merkintäkielellä*. Noudettu osoitteesta [https://publications.theseus.fi/bitstream/handle/10024/63970/Kuusinen\\_Niko.pdf?sequence=1](https://publications.theseus.fi/bitstream/handle/10024/63970/Kuusinen_Niko.pdf?sequence=1)

Klátik. (2013). *GREENRIDERS Windows Phone application*. Retrieved 11 26, 2013, from [http://www.theseus.fi/bitstream/handle/10024/61074/GRWP\\_BT\\_klatik.pdf?sequence=1](http://www.theseus.fi/bitstream/handle/10024/61074/GRWP_BT_klatik.pdf?sequence=1)

Lindholm. (2012). *Windows Phone 7 –sovelluskehityksen perusteet ja yksinkertaisen Twittersovelluksen kehitys*. Retrieved from <http://urn.fi/URN:NBN:fi:amk-2012112115816>

MSDN. (2013, 11 26). *Windows Phone Emulator*. Retrieved from <http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff402563%28v=vs.105%29.aspx>

MSDN. (n.d.). *App activation and deactivation for Windows Phone 2013*. Retrieved 11 26, 2013, from <http://msdn.microsoft.com/en-us/library/windowsphone/devlop/ff967547%28v=vs.105%29.aspx>

MSDN. (2013, 11 26). *Getting started with developing for Windows Phone*. Retrieved from [http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff402529%28v=vs.105%29.aspx#BKMK\\_1Getsetuptup](http://msdn.microsoft.com/en-us/library/windowsphone/develop/ff402529%28v=vs.105%29.aspx#BKMK_1Getsetuptup)

Pahkala. (2012). *Introduction to Windows Phone 8*. Retrieved 11 26, 2013, from <https://publications.theseus.fi/handle/10024/51260>

Peltoniemi. (2011). *Windows Phone 7 -sovelluskehitys*. Retrieved from <http://urn.fi/URN:NBN:fi:amk-201105137804>

Timonen. (2013). *Windows Phonen ja HTML5:n sovelluskehitys*. Retrieved 11 26, 2013, from [http://www.theseus.fi/bitstream/handle/10024/61407/Timonen\\_Olli.pdf?sequence=1](http://www.theseus.fi/bitstream/handle/10024/61407/Timonen_Olli.pdf?sequence=1)

The Visual Studio Blog. (2013). *Introducing Windows Phone SDK 8.0*. Retrieved 11 26, 2013, from <http://blogs.msdn.com/b/visualstudio/archive/2012/10/30/introducing-windows-phone-sdk-8-0.aspx>