



**SAVONIA**

# **Innovaatioprosessi mobiilisovelluskehityksessä**

**Pirinen Juha**

Opinnäytetyö

---

**Ylempi ammattikorkeakoulututkinto**



Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Hyvinvointiteknologian koulutusohjelma	
Työn tekijä(t) Juha Pirinen	
Työn nimi Innovaatioprosessi mobiilisovelluskehityksessä	
Päiväys 26.11.2012	Sivumäärä/Liitteet 65/6
Ohjaaja(t) lehtori Jussi Koistinen, lehtori Sami Lahti	
Toimeksiantaja/Yhteistyökumppani(t) KajaPro Oy	
Tiivistelmä <p>Tämän opinnäytetyön tarkoituksena oli kehittää uusi innovaatioprosessi KajaPro Oy:lle. Kehitetyn innovaatioprosessin avulla tuli tutkia Windows Phone -alustan tarjoamia mahdollisuuksia yrityksen ohjelmistokehitykselle. Tutkimusta varten tuli luoda esimerkkisovellus, joka esittelee Windows Phonen perustoimintoja. Sovelluksen tuli olla kaupallisesti hyödynnettävissä.</p> <p>Ennen päivitetyn prosessin esittämistä täytyi tutustua yleisen innovaatioprosessin teoriaan sekä pienen ohjelmistoyrityksen erityisvaatimuksiin innovaatioprosessin kannalta. Koska yrityksen innovaatioprosessi liittyy mobiilisovellusten suunnitteluun ja kehittämiseen, täytyi perehtyä myös yleisimpiin mobiililaitteisiin ja mobiilikäyttöjärjestelmiin sekä näiden ohjelmointiin liittyviin asioihin. Erään määritelmän mukaan ideasta tulee innovaatio vasta, kun se tuottaa taloudellista hyötyä. Tämän vuoksi täytyi tutkia sovellusten jakeluun ja markkinointiin liittyviä asioita.</p> <p>Innovaatioprosessia testattiin käytännössä työn aikana. Testauksen aikana kehitettiin sovellus Windows Phone -alustalle. Sovelluksen toteutuksen aikana kerrattiin sovelluskehitykseen käytettävien työkalujen toimintaa sekä perehdyttiin Windows Phone -alustaan liittyviin ohjelmointitekniikoihin. Sovellus toteutettiin käyttäen Microsoftin Visual Studio 2010 for Windows Phonea ja Expression Blend 4:ää.</p> <p>Työn tuloksena syntyi esitys yrityksen uudeksi innovaatioprosessiksi. Innovaatioprosessia aletaan ottaa käyttöön mahdollisuuksien mukaan työn valmistumisen jälkeen. Lisäksi syntyi sovellus, jolla Windows Phonen tarjoamia mahdollisuuksia on mahdollista esitellä. Sovellus on tarkoitus laittaa myyntiin Microsoftin kauppapaikkaan viimeistelyn jälkeen. Innovaatioprosessin testaamisen aikana havaittiin, että yrityksen kannattaa odottaa Windows Phone 8:n julkaisua, ennen kuin Windows Phone -sovelluksia aletaan kehittää täysipainoisesti.</p>	
Avainsanat innovaatio, innovaatioprosessi, mobiilikäyttöjärjestelmä, ohjelmistokehitys, Windows Phone	
julkinen	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme of Health and Welfare Technology			
Author(s) Juha Pirinen			
Title of Thesis Innovation Process in Mobile Application Development			
Date	26 November 2012	Pages/Appendices	65/6
Supervisor(s) Mr Jussi Koistinen, Lecturer; Mr Sami Lahti, Lecturer			
Client Organisation/Partners KajaPro Ltd.			
<p><b>Abstract</b></p> <p>The purpose of this thesis was to update the innovation process of KajaPro Ltd. The secondary purpose was to study the possibilities provided by the Windows Phone platform using the updated innovation process. A Windows Phone based application will be used to demonstrate the possibilities of the platform. The application should be commercially profitable.</p> <p>It is important to utilise the whole innovation potential of the company. This is even more important for the modern software company. If the company has an innovation process that helps to utilise the potential, it will be a huge benefit in commercial markets.</p> <p>The theory of a general innovation process and the introduction of the presented innovation process had a major role in this thesis. Before it was possible to introduce an updated innovation process for the company, the theory of general innovation process and special needs of a small software company had to be studied. Monitoring the general development of mobile platforms is an important part of the innovation process of the company. Because of this the most popular mobile operating systems and devices were studied together with the software development for these platforms. An idea really becomes an innovation after it starts to give commercial profit. For this, the marketing of mobile applications in general and the marketing channels of three platform providers were studied and introduced.</p> <p>The presented innovation process was briefly tested. An application for the Windows Phone platform was developed during this test. During development, the usage of software development tools and programming techniques for Windows Phone were studied. Visual Studio 2010 Express for Windows Phone and Expression Blend 4 by Microsoft were used to develop the application.</p> <p>Both parts of the thesis were successful. The result from the first part was an updated innovation process for the company. The result of the second part was an application for Windows Phone 7. With the application it was possible to demonstrate the functionalities provided by the platform. The application will be finalized and moved to Microsoft Marketplace later. As the innovation process was tested, it showed that the publication of Windows Phone 8 should be waited before adopting Windows Phone platform for the company.</p>			
<p><b>Keywords</b> innovation, innovation process, mobile operating system, software development, Windows Phone public</p>			

## ALKUSANAT

Haluan kiittää KajaPro Oy:tä opinnäytetyön aiheen ideoinnista ja kiinnostuksesta työn edistymistä kohtaan. Erityiskiitos kuuluu ohjelmistosuunnittelija Mika Keräselle, työn ohjaajalle KajaPro:n puolelta. Hänen kanssaan oli mahdollista keskustella työn etenemisestä ja muista työhön liittyvistä asioista aina, kun oli tarpeen. Kiitos kuuluu myös opettaja Jussi Koistiselle työn asiantuntevasta ohjauksesta.

Kajaanissa 19.11.2012

Juha Pirinen

## SISÄLTÖ

1	JOHDANTO .....	8
2	MOBIILIOHJELMOINTI JA -KÄYTTÖJÄRJESTELMÄT .....	9
2.1	Mobiililaitteet .....	9
2.2	Yleisimmät mobiilikäyttöjärjestelmät .....	10
2.3	Android .....	11
2.4	iOS .....	11
2.5	Symbian.....	13
2.6	Windows Phone 7 .....	13
2.6.1	Työkalut .....	14
2.6.2	Sovelluskehukset ja käytettävät ohjelmointikielet.....	15
2.6.3	Sensorit ja kommunikointi ulkoisten laitteiden kanssa .....	15
2.6.4	Nokian julkistusten vaikutus .....	16
2.7	Muut mobiilikäyttöjärjestelmät .....	17
2.8	HTML5 ja JavaScript mobiiliohjelmoinnissa .....	18
3	INNOVAATIOPROSESSI.....	19
3.1	Innovaation määritelmä.....	19
3.2	Innovoinnin ongelmat yrityksen sisällä .....	20
3.3	Yleinen innovaatioprosessi .....	21
3.3.1	Innovaatioprosessin alkupää.....	22
3.3.1.1	Mahdollisuuksien tunnistaminen .....	22
3.3.1.2	Ideointi.....	24
3.3.1.3	Ideoiden kehittäminen.....	26
3.3.1.4	Ideoiden arvioiminen.....	27
3.3.2	Palkitsemiskäytännöt .....	29
3.3.3	Konseptointi.....	29
3.3.3.1	Konseptoinnin määritelmä .....	29
3.3.3.2	Konseptoinnin kategoriat .....	30
3.3.3.3	Visioiva konseptointi .....	31
3.3.3.4	Kehittävä konseptointi.....	32
3.3.3.5	Määrittelevä konseptointi .....	32
3.3.3.6	Ratkaiseva konseptointi.....	33
3.3.3.7	Yhteenveto konseptoinnista.....	33

3.3.4	Tuoteprosessi .....	34
3.3.5	Innovaation kaupallistaminen ja markkinointi.....	36
4	MOBIILISOVELLUSTEN MARKKINOINTI JA JAKELU .....	38
4.1	Yleistä mobiilisovellusten markkinoinnista .....	38
4.2	Apple.....	39
4.3	Microsoft.....	40
4.4	Google.....	41
5	KAJAPRO OY:N INNOVAATIOPROSESSI .....	42
5.1	Innovaatioprosessi pienessä yrityksessä .....	42
5.2	Ohjelmistokehityksen innovaatioprosessi.....	44
5.2.1	Prosessin alku.....	44
5.2.2	Vesiputousmalli.....	45
5.2.3	Ketterät menetelmät.....	46
5.3	Eesitys innovaatioprosessiksi .....	47
5.3.1	Innovaatioprosessin alkupää .....	48
5.3.1.1	Mahdollisuuksien tunnistaminen .....	48
5.3.1.2	Ideointi.....	50
5.3.1.3	Ideoiden jatkokehittäminen .....	52
5.3.1.4	Ideoiden arviointi.....	53
5.3.2	Konseptointi .....	53
5.3.3	Tuoteprosessi .....	55
5.3.4	Palkitsemiskäytännöt.....	55
6	INNOVAATIOPROSESSIN TESTAAMINEN KÄYTÄNNÖSSÄ.....	56
6.1	Mahdollisuuksien tunnistaminen ja ideointi.....	56
6.2	HTML5:n tarjoamat mahdollisuudet.....	57
6.3	Sovellus .....	57
6.4	Innovaatioprosessin tulokset.....	58
7	YHTEENVETO JA POHDINTA.....	59
	LÄHTEET .....	61

## LIITTEET

Liite 1 Esimerkkejä Windows Phone -sovelluskehitykseen käytettävistä työkaluista

Liite 2 Toteutetun Windows Phone -sovelluksen näyttökuvia

## 1 JOHDANTO

KajaPro on Kajaanista lähtöisin oleva ohjelmistoalan yritys, joka työllistää noin 20 henkeä. Yrityksellä on tällä hetkellä toimipisteet Kajaanissa ja Oulussa. Yrityksen työntekijät toimivat pääasiassa asiakkaiden projekteissa alihankkijoina mutta yrityksellä on myös omaa tuotekehitystä.

Yrityksen omaa tuotekehitystä tukemaan tarvitaan toimiva innovaatioprosessi. Innovaatioprosessin avulla on mahdollista saada yrityksen työntekijöiden innovointipotentiaali paremmin käyttöön. Tämän opinnäytetyön tavoitteena on luoda esitys yrityksen uudeksi innovaatioprosessiksi. Lisäksi tulee tutkia Windows Phone -alustan sopivuutta yrityksen käyttöön.

Työ on jaettu kahteen osaan. Ensimmäisessä osassa luodaan esitys yrityksen uudeksi innovaatioprosessiksi. Tätä osiota varten tuli tutustua omatoimisesti innovaatioprosessin teoriaan, koska se ei ollut kaikilta osiltaan tuttua eikä innovaatioprosessin teorian opiskelu kuulunut opetussuunnitelmaan. Toisessa osassa innovaatioprosessia testataan ja sen avulla tutkitaan Windows Phone -alustan tarjoamia mahdollisuuksia yrityksen tuotekehitykselle sekä pyritään luomaan sovellus Windows Phonelle.

Koska KajaPro:n innovaatioprosessissa täytyy seurata useita erilaisia mobiilialustoja, raportin aluksi esitellään lyhyesti yleisimmät mobiililaitteiden tyypit ja mobiilikäyttöjärjestelmät sekä niiden ohjelmistokehitykseen liittyvät asiat. Tämän jälkeen esitellään innovaatioprosessin teoria. Koska sovellusten markkinointi ja jakelu liittyvät läheisesti KajaPro:n innovaatioprosessiin, on raportissa käsitelty lyhyesti mobiilisovellusten markkinointia ja kolmen mobiilialustan tarjoajan jakelukanavat. Lopuksi kuvataan innovaatioprosessin testaus käytännössä. Tässä yhteydessä käsitellään sovelluksen toteuttamiseen liittyvät asiat pääpiirteissään. Sovelluksen toteutusta ei ole kuvattu yksityiskohtaisesti, koska se ei ollut työn kannalta oleellista.



## 2 MOBIILIOHJELMOINTI JA -KÄYTTÖJÄRJESTELMÄT

Mobiiliohjelmoinnilla tarkoitetaan mobiilien päätelaitteiden ohjelmointia. Mobiiliohjelmoinnin käsite on hyvin laaja ja se käsittää monia erilaisia ohjelmointiympäristöjä ja -kieliä. Kun sovellusta tehdään jollekin tietylle päätelaitteelle ja käyttöjärjestelmälle, tarkoitetaan natiivin sovelluksen kehittämistä. Natiivien sovelluksien ohjelmointikielinä käytetään esimerkiksi Javaa, C#:a tai C++:aa. Sovellukset voidaan toteuttaa myös niin, että ne toimivat usealla eri alustalla. Tällaiset sovellukset ovat yleensä web-pohjaisia ja niiden ohjelmoinnissa voidaan käyttää esimerkiksi HTML5-kieltä. (Blueberry consultants 2012.)

### 2.1 Mobiililaitteet

Ennen mobiililaitteiden käyttöjärjestelmien käsittelyä on tarkoituksenmukaista avata mobiililaitteen käsitettä. Jukka Korpelan (2008) Pienehkössä sivistyssanakirjassa mobiili-sana määritellään seuraavasti: liikkuva, liikuteltava, siirrettävä, langaton.

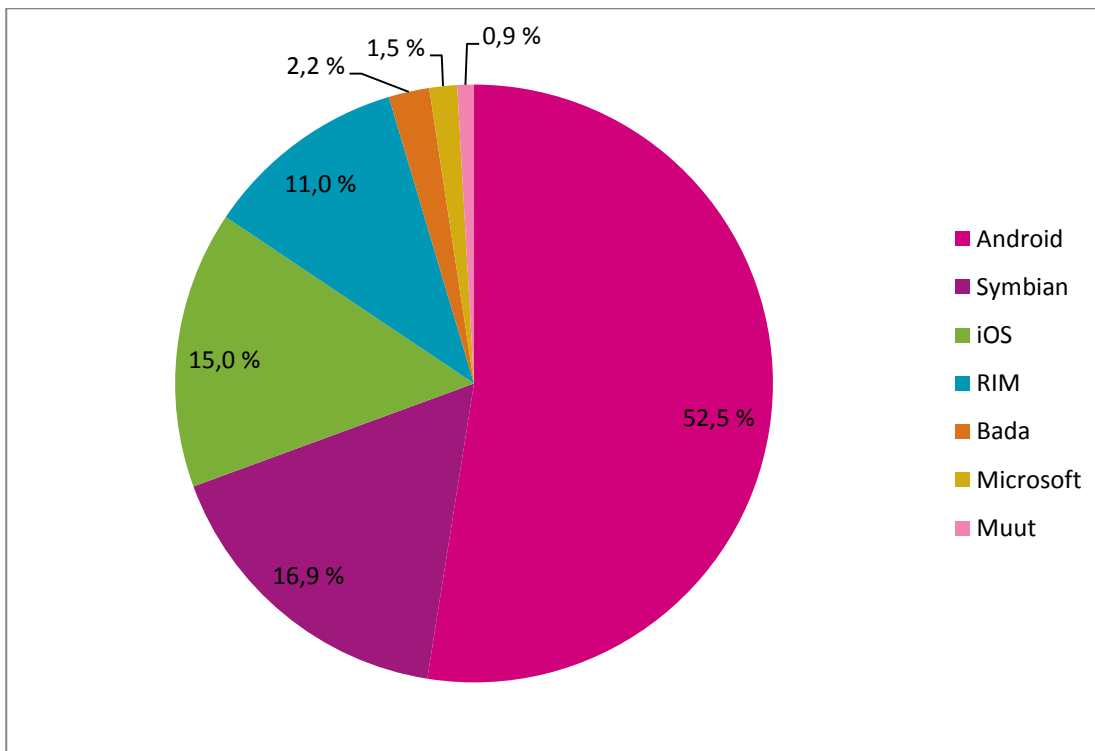
Mobiililaitteella tarkoitetaan yleensä pienikokoista, kädessä pidettävää, tietojenkäsittelyyn pystyvää laitetta. Laitteissa on yleensä joko kosketusnäyttö tai perinteinen näyttö ja mahdollisesti erillinen näppäimistö. Tällaisia laitteita ovat esimerkiksi perinteiset matkapuhelimet, älypuhelimet, kämmentietokoneet ja GPS-navigaattorit. Lisäksi voidaan mainita pääasiassa tekstin lukuun tarkoitettut e-reader-laitteet sekä tablet-tietokoneet. Yleisesti voidaan sanoa, että mobiililaitteella voidaan vastaanottaa, käsitellä ja lähettää tietoa paikasta riippumatta. (Viswanathan 2012; Tuominen 2010, 8.)

On olemassa myös muunlaisia sulautettuja mobiililaitteita, jotka eivät kuulu edellä mainittujen laitteiden joukkoon, vaikka sisältävätkin samoja toimintoja ja ominaisuuksia. Nämä laitteet on yleensä suunniteltu jotain tiettyä toimintoa varten esimerkiksi teollisuuden käyttöön. Tässä työssä keskitytään tavallisten kuluttajien käytössä oleviin laitteisiin, joten em. laitteita ei käsitellä tarkemmin.

## 2.2 Yleisimmät mobiilikäyttöjärjestelmät

Mobiilikäyttöjärjestelmällä tarkoitetaan nykyisin yleensä luvussa 2.1 esiteltyihin, yleisessä käytössä oleviin mobiililaitteisiin tarkoitetusta käyttöjärjestelmästä. Luvussa 2.1 mainittujen erikoiskäyttöön suunniteltujen laitteiden käyttöjärjestelmiä ovat esimerkiksi uLinux, eCos ja µCos-II. Tässä luvussa esitellään yleisessä käytössä oleviin laitteisiin tarkoitettuja käyttöjärjestelmiä. Erikoiskäyttöjärjestelmiä ei tässä työssä käsitellä tarkemmin.

Vuoden 2011 lopussa yleisimmät käytössä olevat mobiilikäyttöjärjestelmät olivat Googlen Android, Applen iOS ja Nokian poistumassa oleva Symbian. Tänä ajankohdalla Microsoftin Windows Phonella ei vielä ollut mainittavaa markkinaosuutta. Kuviossa 1 on esitetty mobiilikäyttöjärjestelmien markkinaosuuksien tilanne vuoden 2011 kolmannen neljänneksen lopussa.



KUVIO 1. Mobiilikäyttöjärjestelmien markkinaosuudet vuoden 2011 kolmannen neljänneksen lopussa (Gartner 2011).

## 2.3 Android

Android on ohjelmistopino mobiililaitteille. Siihen sisältyy käyttöliittymä, middleware ja avainsovellukset toimintojen käyttämiseen. Ensimmäinen versio Androidista julkaistiin vuonna 2007. Sitä ylläpitää ja kehittää Googlen johtama Open Handset Alliance. (Google 2012, Developers.)

Androidin ohjelmistokehitys tehdään pääosin Android SDK:lla. SDK:ssa ohjelmointikielenä käytetään Javaa. Sovelluskehitystä varten on olemassa myös NDK, jossa on mahdollista käyttää natiiveja ohjelmointikieliä (C ja C++). NDK tarjoaa ohjelmointikirjastot ko. ohjelmointikielille. Kirjastojen avulla voidaan käsitellä esimerkiksi käyttäjän syötteitä, laitteistopohjaisia sensoreita ja sovellusresursseja. Natiivia koodia kirjoittamalla on periaatteessa mahdollista kirjoittaa sovelluksia, jotka toimivat tehokkaammin kuin SDK:lla tehdyt sovellukset. NDK toimii yhdessä SDK:n kanssa. (Google 2012, Developers.)

Natiivikoodia käyttävien sovellusten kehittämiseen täytyy olla oikea tarve ennen kuin niiden kehittäminen kannattaa. Automaattisesti natiivikoodi ei tehosta sovelluksen suoritusta, mutta se lisää sovelluksen kompleksisuutta. Vanhojen C/C++-kielillä kirjoitettujen sovellusten siirtäminen Androidille voi olla perusteltu syy natiivin koodin kirjoittamiseen sovellusten tehostamisen lisäksi. (Google 2012, Developers.)

Android NDK:n käytöstä on hyötyä, jos samaa sovellusta halutaan käyttää usealla eri alustalla. Pelkkää Javaa käytettäessä ongelmana on, että koko sovellus joudutaan kirjoittamaan pääosin uudestaan, jos se halutaan toimimaan toisella alustalla. Useat alustat, kuten iOS, Bada ja Symbian, tukevat C/C++-kieliä, joten pelkästään alusta-kohtaiset osiot sovelluksesta täytyy ohjelmoida uudestaan.

## 2.4 iOS

iOS on Applen kehittämä käyttöjärjestelmä, jota se käyttää omissa mobiililaitteissaan (iPhone, iPod touch, iPod Nano ja iPad). Käyttöjärjestelmän ensimmäinen versio julkaistiin vuonna 2007. Käyttöjärjestelmä nimettiin iPhone OS:ksi vuonna 2008 ja nimi muutettiin iOS:ksi 2010. (Bohn 2011.)

Sovelluskehittäjät liittyvät Applen iOS Developer Program -nimiseen ohjelmaan. Ohjelman vuotuinen jäsenmaksu on 99 dollaria. Ohjelmistokehitys iOS-käyttöjärjestelmälle suoritetaan iOS SDK:lla. iOS SDK sisältää seuraavat työkalut:

- Xcode
  - Kehitysympäristö, jossa on projektin hallinta, koodieditori ja graafinen debuggeri
- iOS Simulator
  - iPhone ja iPad simulaattorit, joilla voi ajaa, debugata ja testata kehitettäviä sovelluksia
- Instruments
  - Työkalun avulla voidaan kerätä, esittää ja vertailla kehitettävän sovelluksen suorituskykytietoja graafisesti ja reaaliaikaisesti sovelluksen optimointia varten
- Interface Builder
  - Käyttöliittymän kehitystyökalu, jonka avulla voidaan tehdä sovelluksen käyttöliittymät ”drag and drop” -menetelmällä (Apple 2012.)

Sovellukset iOS:lle ohjelmoidaan pääasiassa käyttäen Objective-C-ohjelmointikieltä. Objective-C on käytännössä olio-pohjainen laajennus ANSI C:hen. Olio-laajennus perustuu pääasiassa Smalltalkiin, joka on yksi ensimmäisistä olio-ohjelmointikielistä. Koska Objective-C on laajennus C-kieleen, on sekä C-kielen että C++-kielen käyttö ohjelmoinnissa mahdollista. (Apple 2012.)

iOS:lle on mahdollista kehittää sovelluksia myös HTML5:llä. Näin toteutetuista sovelluksista on mahdollista tehdä natiivisovellusten kaltaisia ja niitä voidaan käyttää, vaikkei yhteyttä verkkoon olisikaan saatavilla. (Kissinger 2012.)

Tarkemmin HTML5:n käytöstä mobiiliohjelmoinnissa on kerrottu luvussa 2.8.

## 2.5 Symbian

Symbian on älypuhelimille tarkoitettu käyttöjärjestelmä. Sen eri versiot ovat käytössä pääasiassa Nokian mobiililaitteissa. Kirjoitushetkellä Symbiania ylläpitää Accenture. Symbian-käyttöjärjestelmän edeltäjän Symbian OS:n on alun perin kehittänyt Symbian Ltd. Yhtiö perustettiin 24.6.1998 Ericssonin, Nokian, Motorolan ja Psionin yhteistyönä. (Nokia Developer Wiki 2012.)

Vuodesta 2010 alkaen Symbianin sovelluskehitykseen on pääasiallisesti käytetty standardia C++-kieltä ja Qt SDK:ta. Kehitysympäristönä on mahdollista käyttää joko Qt Creatoria tai Carbidea. Kehitysympäristön mukana tulee simulaattori, jolla sovellusta on mahdollista testata. Edellä mainittujen tekniikoiden lisäksi Symbian pohjaisia laitteita voidaan ohjelmoida käyttäen Pythonia, Java ME:tä, Flash Lite:ä, Ruby:a, .NET:iä, Web Runtime (WRT) Widgetejä ja standardia C/C++:aa. (Nokia Developer Wiki 2012.)

Nokia ilmoitti 11.2.2011, että se siirtyy käyttämään Microsoftin Windows Phonea pääasiallisena käyttöjärjestelmänä tulevissa älypuhelimissaan. Symbian ei kuitenkaan Nokian mukaan häviä kokonaan, vaan jää vielä käyttöön halvemman hintaluokan puhelimiin. (Jaakola 2011.)

Windows Phone 7 -käyttöjärjestelmästä on kerrottu tarkemmin luvussa 2.6.

## 2.6 Windows Phone 7

Windows Phone 7 on Microsoftin kehittämä mobiilikäyttöjärjestelmä. Se on kehitetty Microsoftin aiemman käyttöjärjestelmän, Windows Mobilen, seuraajaksi. Windows Phone 7 ei kuitenkaan ole yhteensopiva Windows Mobilen kanssa. Windows Mobilea käytäviä laitteita ei myöskään ole mahdollista päivittää käyttämään Windows Phonea. (Amy 2010.)

Työn aikana tarkemmin tutkittava alusta on Windows Phone 7. Tämän vuoksi Windows Phone -sovelluskehitystä kuvaillaan tarkemmin kuin sovelluskehitystä muille alustoille.

### 2.6.1 Työkalut

Windows Phone -sovellukset kehitetään käyttäen Microsoftin työkaluja ja tekniikoita. Näihin kuuluu Visual Studio, Expression Blend, Silverlight ja XNA. Sovelluskehitykseen tarvittavat komponentit on kasattu Windows Phone SDK -nimiseen pakettiin. Windows Phone SDK sisältää Visual Studio 2010 Express for Windows Phonen, Windows Phone emulaattorin, Expression Blend for Windows Phonen ja XNA Game Studion. Lisäksi SDK:n mukana toimitetaan sovellusesimerkkejä ja dokumentaatiota. Kirjoitushetkellä SDK:sta on käytössä versio 7.1. (Microsoft 2012, MSDN Library.)

Itse sovelluskehitys tehdään käyttäen Visual Studiota. Visual Studio on Microsoftin integroitu sovelluskehitysympäristö, joka sisältää tarvittavat toiminnot sovellusten kehittämiseen Microsoftin alustoille Windows ympäristössä. Windows Phone SDK sisältää ilmaisen version Visual Studiosta. (Microsoft 2012, Visual Studio.)

Windows Phone emulaattorin avulla kehitettävää sovellusta voidaan testata ja siitä voidaan etsiä virheitä (debuggaus). Windows Phone emulaattori on työpöytäsovellus, joka on Windows Phone SDK:ssa integroitu osaksi Visual Studiota ja Expression Blendiä. Sillä voidaan emuloida Windows Phone 7 ja 7.1 laitteita. Microsoftin mukaan emulaattorin suorituskyky ja ominaisuudet on suunniteltu vastaamaan todellista laitetta ja esimerkiksi sensoreiden ja sovelluksen välistä tiedonsiirtoa voidaan testata kuten oikealla laitteella. (Microsoft 2012, MSDN Library.)

Expression Blend on työkalu käyttöliittymien suunnitteluun. Sen avulla voidaan luoda käyttöliittymiä, joissa yhdistellään web- ja työpöytäsovellusten ominaisuuksia. Käyttöliittymät voidaan suunnitella joko graafisessa tilassa tai muokkaamalla XAML-tiedostoja, jotka kuvaavat käyttöliittymän tekstimuodossa. Yleensä sovelluskehityksen aikana täytyy käyttää molempia tiloja, että käyttöliittymästä saadaan toimiva. Expression Blend on WYSIWYG-tyylinen työkalu. Tämä tarkoittaa sitä, että lopullinen käyttöliittymä on samannäköinen kuin graafisessa suunnittelutilassa voidaan nähdä. (Microsoft 2012, Expression.)

Liitteessä 1 on esitetty kuvia käytettävistä työkaluista.

### 2.6.2 Sovelluskehukset ja käytettävät ohjelmointikiel

Windows Phone sovellusalustassa on kaksi sovelluskehystä (framework), joita voidaan käyttää kehitettäessä sovelluksia: Silverlight ja XNA. Silverlight on tarkoitettu perinteisten business-sovellusten kehukseksi. Silverlightin avulla voidaan kehittää tapahtumapohjaisia sovelluksia joilla on RIA-tyylinen käyttöliittymä. XNA on tarkoitettu pääasiassa pelien kehittämiseen. Se sisältää useita ohjelmointirajapintoja pelien kehittämistä varten. 2D-pohjaiset sprite-rajapinnat tukevat esimerkiksi skaalausta, venytystä ja kuvan suodatusta. 3D-grafiikkarajapinnat tukevat 3D-geometriaa, tekstuureja ja valaistukseen liittyviä asioita. XNA:n käyttö nopeuttaa tarvittaessa pelin siirtämistä toiselle Microsoftin alustalle. Muita alustoja kirjoitushetkellä ovat Xbox 360, Zune HD ja Windows 7. Windows Phone 7.5:n myötä Silverlight ja XNA-sovelluksia on mahdollista yhdistää keskenään. (Microsoft 2012, MSDN Library.)

Ohjelmointikielenä Windows Phone sovelluksille käytetään C#:ia tai Visual Basicia (Microsoft 2012, MSDN Library). Ongelmallista näiden kielten käytössä on, että niillä kirjoitettu sovellus joudutaan kirjoittamaan käytännössä kokonaan uudestaan, jos se halutaan siirtää toiselle alustalle. Tämän vuoksi ohjelmistojen kehittäjät halusivat käyttöönsä Windows Phone NDK:n, jonka avulla sovelluksissa voitaisiin käyttää C/C++ -kieliä. Windows Phone NDK:n saapumisesta kehittäjille on liikkunut huhuja jo pitkään, mutta ainakaan kirjoitushetkellä sitä ei ole saatavilla.

### 2.6.3 Sensorit ja kommunikointi ulkoisten laitteiden kanssa

Windows Phone SDK:ssa on ohjelmointirajapinnat monille laitteissa oleville sensoreille. Rajapinnat löytyvät esimerkiksi kiihtyvyysanturille, kompassille, gyroskoopille, mikrofonille ja monikosketuksen tunnistukselle. (Microsoft 2012, MSDN Library.)

Ohjelmointikielien lisäksi sovelluskehittäjän kannalta on ongelmallista, että Windows Phone SDK:ssa ei ole Bluetooth-ohjelmointirajapintaa. Tämä aiheuttaa sen, että SPP tai pakettipohjaista tiedonsiirtoa Bluetooth-yhteyttä käyttävien oheislaitteiden kanssa ei ole mahdollista toteuttaa. Bluetooth-ohjelmointirajapinta on kuitenkin olemassa. Sen käyttöön on tähän mennessä annettu lupa pelkästään Microsoftin yhteistyökumppaneille. Rajapinnan julkaisemisesta yleiseen käyttöön on liikkunut paljon huhuja, mutta virallista tietoa julkaisemisesta ei ole.

Yksi mahdollisuus ulkoisten laitteiden kanssa kommunikointiin on sokettipohjaisen tiedonsiirron käyttäminen. Sokettipohjaisessa tiedonsiirrossa tarvitaan verkkoyhteys tietoa siirtävien kohteiden välillä. Windows Phone 7 mahdollistaa UDP ja TCP/IP-sokettien käytön. (Microsoft 2012, MSDN Library.)

Sokettipohjainen tiedonsiirto on käyttökelpoinen vaihtoehto, jos ulkoinen laite on tehty tai voidaan tehdä niin, että se on mahdollista liittää verkkoon. Sokettiyhteyttä voidaan käyttää esimerkiksi WLAN, 3G ja langallisen verkkoyhteyden kautta. Ulkoisten laitteiden kanssa kommunikointiin erityisesti kuluttajasovelluksissa, kuten sydämen sykkeen seurantaan kuntoilun aikana, sokettiyhteydet eivät ole käytännöllisin vaihtoehto. Teollisuuskäytössä esimerkiksi sensoriverkon tuottamien tietojen seurantaan yhteys sen sijaan voi olla järkevästi hyödynnettävissä.

#### 2.6.4 Nokian julkistusten vaikutus

Kuten luvussa 2.5 mainittiin, Nokia tulee käyttämään uusissa älypuhelimissaan Windows Phone -käyttäjärjestelmää. Jos Nokia onnistuu saavuttamaan merkittävän markkina-aseman näillä puhelimilla, on sillä suuri vaikutus Windows Phone 7:n levinneisyyteen ja siihen, että ohjelmistokehittäjät kiinnostuisivat alustasta. Windows Phonen levinneisyys oli vuoden 2011 lopussa pientä, kuten luvussa 2.2 kerrottiin. Ensimmäiset Windows Phone 7 -puhelimensa (Lumia 800 ja 710) Nokia julkaisi 26.10.2011. Myyntiin puhelimet luvattiin Euroopassa marraskuun 2011 aikana ja tässä myös onnistuttiin. (Lappalainen 2011.)

Nokian ykkösmalli kirjoitushetkellä on Lumia 900, joka julkaistiin Yhdysvaltojen markkinoille Las Vegasin CES-messuilla 9.1.2012. Julkistetussa mallissa on mahdollisuus käyttää 4G-tekniikan verkkoa, joka on jo yleistymässä Yhdysvalloissa. Yhdysvaltojen markkinoille Lumia 900 tuli huhtikuun 2012 alussa ja Eurooppaan huhtikuun 2012 lopussa. Euroopan markkinoille tullut malli ei sisällä 4G-tekniikkaa (Kotilainen 2012; Lehto 2012).



Lumian myyntitulokset eri maissa ovat olleet vaihtelevia. Yhdysvaltojen markkinoilla puhelinten myynti on Nokian mukaan ylittänyt odotukset, mutta Euroopassa ei vielä ole päästy tavoitteisiin. Parantaakseen puhelinmalliensa kysyntää Nokia aikoo lisätä mainontaa. Lisäksi Nokia toi uudet Lumia-mallit markkinoille nopeutetulla aikataululla, ja se pyrkii toteuttamaan avainasiakkaiden toivomia toiminnallisuuksia laiteisiinsa. (Rapeli 2012.)

Kirjoitushetkellä on vielä vaikea sanoa, kuinka suuri vaikutus Nokian puhelimilla on ollut Windows Phonen levinneisyyteen. Lumiat ovat olleet markkinoilla vasta vähän aikaa, eikä niiden todellisia myyntilukuja ole julkaistu. Lumia 900 on saanut pääosin myönteisen vastaanoton (Lehto 2012).

## 2.7 Muut mobiilikäyttöjärjestelmät

Edellisissä luvuissa käsiteltyjen käyttöjärjestelmien lisäksi on olemassa muutamia käyttöjärjestelmiä, jotka ovat mainitsemisen arvoisia.

Kuviossa 1 neljännellä sijalla mainittu RIM (Research In Motion) on kanadalainen vuonna 1984 perustettu yritys. Se toi vuonna 1999 markkinoille ensimmäiset BlackBerry-tuoteperheeseen kuuluvat älypuhelimet. Kirjoitushetkellä BlackBerry-tuoteperheeseen kuuluu sekä tablet-tietokoneita että älypuhelimia. Älypuhelimien käyttöjärjestelmänä toimii BlackBerry OS:n versio 6 tai 7. (RIM 2012.)

Maaliskuun 2012 lopussa RIM ilmoitti, että se pienentää panostustaan kuluttajamarkkinoille ja keskittyy erityisesti yritysasiakkaisiin. Kuluttajamarkkinoilla yritys aikoo keskittyä halvemman hintaluokan puhelimiin. Yritysasiakkaiden joukossa RIM pyrkii saamaan toisille älypuhelinvalmistajille häviämänsä asemat takaisin. Yksi tärkeimmistä asioista tämän tilanteen parantamiseksi tulee olemaan vuoden 2012 loppuun luvattu BlackBerry OS:n version 10 julkaisu, joka on myöhästynyt jo useasti. (BBC 2012.)

Bada on Samsungin kehittämä käyttöjärjestelmä, jota se käyttää Wave-mallisissa älypuhelimissaan. Euroopassa Bada-pohjaiset puhelimet eivät ole saaneet suurta suosiota, mutta vuonna 2011 sen markkinaosuus maailmanlaajuisesti oli Windows Phone -puhelinluokkaa. Badan suurin ongelma on se, etteivät sovelluskehittäjät ole innostuneet siitä. Uutiset keväällä 2012 viittaavat siihen, että Samsung on hiljalleen luopumassa Badan kehityksestä ja on avaamassa sen lähdekoodin. (Vähimaa 2012.)

## 2.8 HTML5 ja JavaScript mobiiliohjelmoinnissa

Yksi vaihtoehto mobiilisovellusten toteuttamiseen on HTML5. Tällä tekniikalla on mahdollista toteuttaa sovelluksia, jotka ovat alustariippumattomia. Vaatimuksena sovelluksen toiminnalle on, että HTML5 on tuettuna laitteen selaimessa. Yleensä HTML5-sovelluksissa käytetään lisäksi JavaScriptiä toteuttamaan tiettyjä toiminnallisuksia.

Ohjelmointikielen ja alustariippumattomuuden lisäksi natiivi- ja HTML5-sovelluksilla on muitakin eroja. Natiivisovelluksella alustan kaikki ominaisuudet on mahdollista ottaa käyttöön ja natiivikäyttöliittymä on lähtökohtaisesti toimivampi kuin yleinen verkkäyttöliittymä. Mahdollisesti suurin HTML5-sovellukseen liittyvä ongelma on, että sovelluksen täydelliseen toimintaan tarvitaan internetyhteys. (Appcelerator 2012, 1.)

HTML5:n avulla on mahdollista toteuttaa myös ns. hybridisovelluksia, jossa yhdistetään natiivisovelluksen ja HTML5-sovelluksen toimintoja. Näin voidaan toteuttaa sovelluksia, josta osa on kohdennettu tietylle alustalle ja alustan spesifisiä toimintoja voidaan ottaa käyttöön, ja osa voidaan käyttää uudestaan toiselle alustalle samankaltaista sovellusta tehtäessä. (Appcelerator 2012, 1.)

Appceleratorin ja IDC:n toteuttaman sovelluskehittäjille kohdennetun kyselyn perusteella HTML5 on noussut sovelluskehittäjien kiinnostuksessa käytännössä kolmanneksi sijalle alustoista, joille aiotaan kehittää sovelluksia vuonna 2012. Suosituin alustana samassa kyselyssä mainitaan iOS ja seuraavalla sijalla on Android. Windows Phone 7 on kyselyssä neljännellä sijalla. (Appcelerator 2012, 3.)

### 3 INNOVAATIOPROSESSI

Tässä luvussa käsitellään innovaatiota ja innovaatioprosessia yleisesti sekä ohjelmistokehityksen näkökulmasta.

#### 3.1 Innovaation määritelmä

Innovaation perinteinen määritelmä on kaupallisesti menestyksekkäs uusi tuote, prosessi, palvelu tai vastaava konsepti (Fogelholm 2009, 13). Kaksi- tai kolmekymmentä vuotta sitten ei kuitenkaan vielä käytetty termiä innovaatio vaan silloin yleisesti käytössä ollut käsite oli luovuus, johon innovaatiot ja keksinnöt kuuluivat (Heikkilä 2010, 183).

Käsitystä innovaatiosta voidaan yrittää selventää innovaation yleisen määritelmän avulla. Suomen kansallisessa innovaatiostrategiassa (2008) innovaatio määritellään seuraavasti:

*Innovaatio ymmärretään hyödynnettynä osaamislähtöisenä kilpailuetuna. Se voi rakentua teknologian soveltamisen ohella esimerkiksi uusille palvelu- ja liiketoimintamalleille, työ- ja toimintatavoille tai tuotekonseptien ja brändien hallinnalle. Tavallisimmin innovaatio syntyy monien osaamisten yhdistelmänä. (Kansallinen innovaatiostrategia 2008, 2.)*

Dundonin (2002, 5 - 6) määritelmässä innovaatio koostuu neljästä siihen liittyvästä elementistä:

1. Luovuus - uusien ideoiden keksiminen
2. Strategia - määrittely, onko idea uusi ja käyttökelpoinen
3. Toteutus - uudeksi ja käyttökelpoiseksi havaitun idean paneminen toimintaan
4. Hyödyllisyys - uuden ja käyttökelpoisen idean toteutuksesta saatavan lisäarvon maksimoiminen

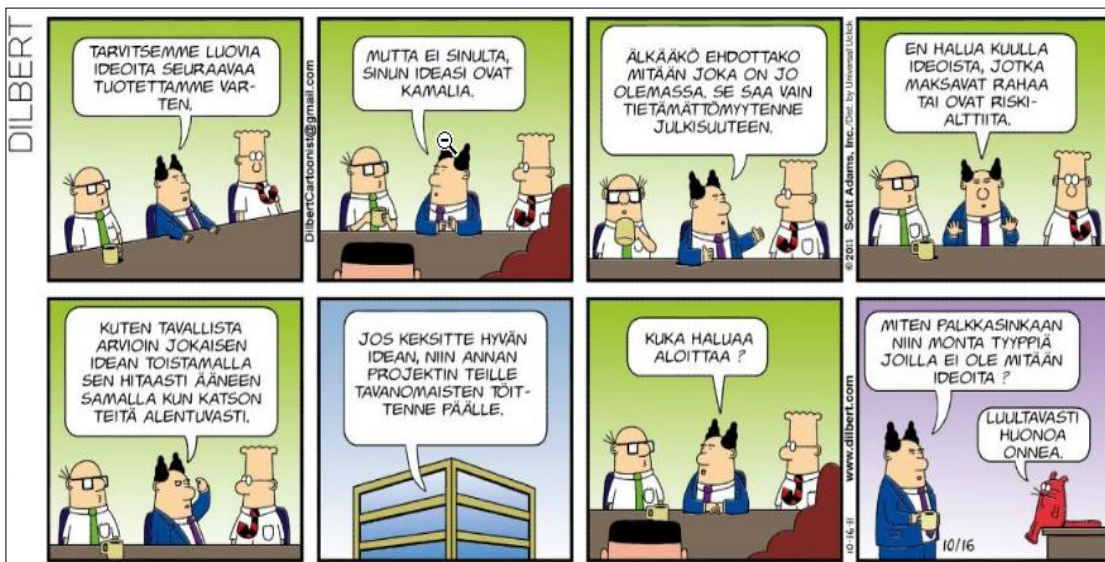
Toteutusvaiheessa monet ideat hylätään joko liian riskialttiina, muutosvastarinnan takia tai siksi, ettei niiden potentiaalista arvoa ole täysin ymmärretty. Idean toteutuksesta saatava lisäarvo voi ilmetä usealla tavalla riippuen innovaation suunnasta: rahallisenä hyötynä, työntekijöiden moraalien paranemisena tai yhteiskunnalle suunnatun tuen lisääntymisenä. (Dundon 2002, 5 - 6.)

Suomessa, joka on globaalisti pieni toimija teknologian alalla, tulisi innovaatioihin kehittämiseen ja kaupallistamiseen panostaa huomattavasti. Näin on pyrittykin tekemään, kuten esimerkiksi kansallisen innovaatiostrategian julkaisemisesta voidaan päätellä. Suomalaisen terveysteknologia-alan yrityksen, Optomedin toimitusjohtaja Seppo Kopsala kuvaa suomalaisten yritysten pyrkimystä innovatiivisuuteen Ojanperän (2012, 5) lehtiartikkelissa *Terveyttä Suomesta* seuraavasti:

*Meillä teknologiayritys ei lähtökohtaisesti perusta liiketoimintaansa olemassa olevien tuotteiden kopioinnille, vaan pyrkii luomaan omanlaisensa tuotteen, jolla on selkeä teknologinen kilpailuetu.*

### 3.2 Innovoinnin ongelmat yrityksen sisällä

Vapaa innovointi voi aiheuttaa vastustusta yrityksissä, koska se haastaa olemassa olevat tavat työskennellä. Yrityksen ilmapiiri voi myös olla sellainen, ettei innovaatioita pääse syntymään tai niitä ei haluta viedä eteenpäin esimerkiksi lisääntyvän työ määrän pelossa. Kuvassa 1 on esitetty esimerkki yrityksestä, jossa mahdolliset innovaatiot eivät tule helposti esille.



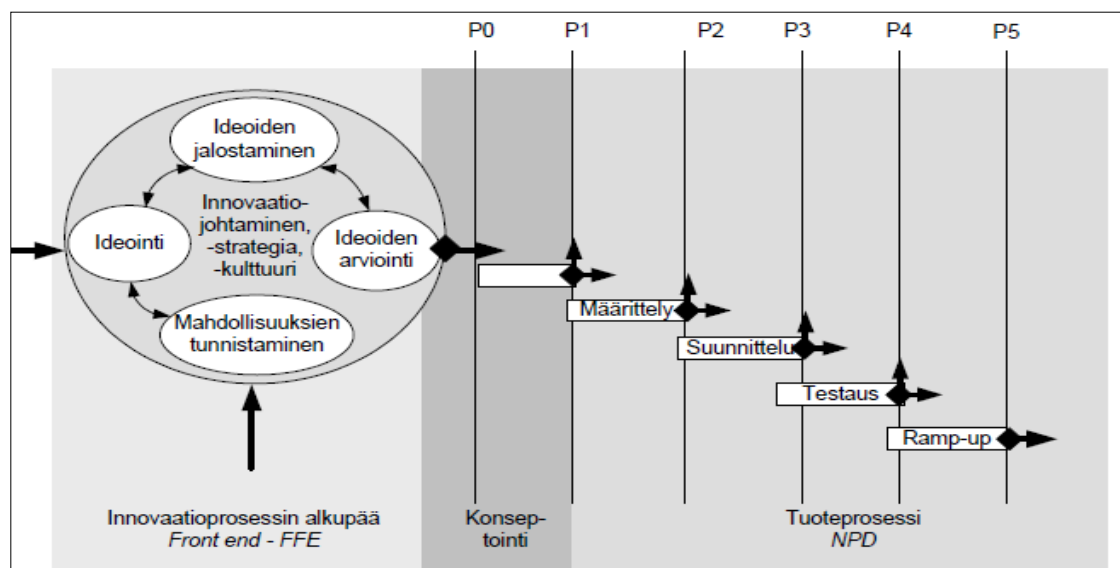
KUVA 1. Esimerkki yrityksen ilmapiiristä, joka ei motivoi työntekijöitä kertomaan mahdollisista innovaatioistaan eteenpäin (Adams, 2011, 19).

### 3.3 Yleinen innovaatioprosessi

Innovaatioprosessi kattaa koko aikavälin ideoiden etsimisestä innovaation syntyyn (Apilo, Taskinen & Salkari 2007, 131). Innovaatioprosessin rakenne riippuu kehitettävän innovaation tyypistä. Yleisen innovaatioprosessin määrittelyä voidaan kuitenkin käyttää hyödyksi, kun halutaan määritellä innovaatioprosessia jollekin erikoistuneelle tapaukselle.

Useimmassa yrityksessä on toimiva tuote- tai tuotekehitysprosessi. Näissä yrityksissä voidaan ajatella, että erillistä innovaatioprosessia ei tarvita. Innovaatioprosessi on kuitenkin laajempi kokonaisuus kuin pelkkä tuote- tai tuotekehitysprosessi. Innovaatioprosessi ei ole pelkästään tuotekehitysosaston asia, vaan siihen liittyy suuri joukko yrityksen muita toimintoja, kuten esimerkiksi myynti, markkinointi, jälkimarkkinointi ja huolto. (Apilo ym. 2007, 132.)

Eri innovaatiotyypeille (esim. tuote-, palvelu- ja prosessi-innovaatio) on mahdollista löytää toisiaan vastaavat vaiheet. Innovaatioprosessi koostuu kahdesta vaiheesta: innovaatioprosessin alkupäästä ja varsinaisesta toteutusvaiheesta. Lisäksi innovaatioprosessiin voi kuulua erillinen tutkimusosuus. Kuvassa 2 on esitetty esimerkki innovaatioprosessin vaiheista. Seuraavissa luvuissa prosessin eri vaiheita on käsitelty tarkemmin. (Apilo ym. 2007, 131 - 132.)



KUVA 2. Esimerkki yleisestä innovaatioprosessista. Innovaatioprosessi koostuu prosessin alkupäästä, konseptoinnista sekä tuoteprosessista. (Apilo & Taskinen 2006, 43.)

### 3.3.1 Innovaatioprosessin alkupää

Innovaatioprosessin alkupäätä voidaan ajatella prosessin haasteellisimpana osana. Sen aikana luodaan innovaation onnistumisen edellytykset ja sen aikana tulee tarkasti miettiä tulevan innovaation lopputulosta. Myöhemmässä vaiheessa mahdollisuudet vaikuttaa lopputulokseen vaikeutuvat ja muutosten hinta nousee huomattavasti korkeammaksi. Myöhemmän vaiheen muutoksia voidaan ennakoida selvittämällä innovaatioon liittyvien teknologioiden, markkinoiden ja asiakkaiden tarpeiden tulevaa kehitystä. Innovaatioprosessin alkupäässä valitaan myös ne ideat, joista innovaatioita aletaan kehittää. (Apilo ym. 2007, 132.)

Innovaatioprosessin alkupää voi pikaisesti tarkasteltuna näyttää kaaosmaiselta tai sumealta. Sitä kuvataankin usein englanninkielisellä termillä fuzzy front end. Käytännössä suurin toiminnallinen ero alkupään ja muun innovaatioprosessin välillä on se, että alkupää on jatkuvaa toimintaa ja loppupää toteutetaan projekteina. Innovaatioprosessin alkupään aikana toteutettavia tehtäviä voidaan tunnistaa, vaikkei alkupäätä voikaan saada tiukasti prosessin muotoon. Alkupään tehtävät voidaan jaotella esimerkiksi seuraavasti: mahdollisuuksien tunnistaminen, ideointi, ideoiden kehittäminen ja ideoiden arvioiminen. (Apilo ym. 2007, 134.)

#### 3.3.1.1 Mahdollisuuksien tunnistaminen

Mahdollisuuksien tunnistaminen voidaan jaotella esimerkiksi seuraaviin kolmeen osaan: asiakastarpeen tunnistaminen, toimialan mahdollisuuksien tunnistaminen ja teknologian mahdollisuuksien tunnistaminen. Mahdollisuuksien tunnistamisen yrityksessä tulisi kuulua mahdollisimman monen henkilön toimenkuvaan, koska eri asemassa olevilla henkilöillä on tehtävänsä ja osaamisensa puolesta paremmat mahdollisuudet ymmärtää ja seurata em. asioita. Organisaation tehtävänä on tukea henkilöstöä näissä tehtävissä. Lisäksi on tärkeää, että eri suunnista mahdollisuuksia tunnistaville henkilöille luodaan tilanteita, joissa he pääsevät keskustelemaan toistensa kanssa. Näin luodaan mahdollisuudet parhaiden ideoiden synnylle. (Apilo ym. 2007, 134.)

On olemassa monia esimerkkejä siitä, että asiakastarpeiden tunnistamisella on suuri vaikutus tuotteen menestymiseen. Perinteisesti asiakastarpeita on tutkittu markkina-tutkimuksilla. Voidaan myös selvittää, miten olemassa oleva tuote ja sen ominaisuudet täyttävät asiakkaiden vaatimukset. Tällaista lähestymistapaa voidaan soveltaa yleensä hitaasti muuttuvilla toimialoilla. (Apilo ym. 2007, 134 - 135.)

Asiakkaalta ei pidä odottaa valmista tuoteideaa tai ratkaisua, vaan kuvaus ongelmasta, joka vaatii ratkaisun. Tämän ongelman pohjalta voidaan yrityksessä alkaa kehittää tuotetta, joka mahdollisesti ratkaisee kyseisen ongelman. Asiakastarpeita voidaan tunnistaa myös käyttämällä apuna käyttäjä- ja käytettävyystudkimusta. Tässä lähestymistavassa etsitään uusia ratkaisuja asiakkaiden määrittelemättömiin tarpeisiin seuraamalla asiakkaan prosessia tai tapaa käyttää olemassa olevia tuotteita. (Apilo ym. 2007, 136.)

Toimialan mahdollisuuksien tunnistaminen on vaikeaa yrityksissä, jotka ovat keskittyneet parantelemaan olemassa olevia tuotteita ja seurailemaan kilpailijoiden liikkeitä. Tällaisissa yrityksissä ei välttämättä nähdä etukäteen pieniäkään toimialaan liittyviä muutoksia, koska niiden tunnistaminen vaatii laajempaa näkemystä ja verkostoitumista sekä oman toimialan että muiden toimialojen sisällä. (Apilo ym. 2007, 138.)

Teknologian mahdollisuuksien tunnistaminen vaatii teknologian kehityksen jatkuvaa seuranta ja se vaatii huomattavasti resursseja. Teknologian kehitystä kannattaa seurata mahdollisimman laaja-alaisesti, koska sitä kautta voidaan löytää uusia mahdollisuuksia sellaisista teknologioista, joissa yrityksellä ei ole aikaisempaa osaamista. (Apilo ym. 2007, 138.)

Teknologian mahdollisuuksien tunnistamiseen on useita keinoja, joista yksi on verkostoituminen yliopistojen ja tutkimuslaitosten kanssa esimerkiksi yhteisrahoitteisten tutkimushankkeiden ja erilaisten projektien sekä tilattujen selvitysten avulla. On myös mahdollista järjestää poikkiteknologisia tai yksittäiseen teknologiaan kohdennettuja teknologiafoorumeita. Jos yrityksen sisältä löytyy eri teknologioiden osaamista, kannattaa nämä osaajat saattaa yhteen ja näin tekemällä mahdollistaa uusien ideoiden syntyminen teknologioiden rajapinnoille. (Apilo ym. 2007, 138 - 139.)

### 3.3.1.2 Ideointi

Ideoinnin täytyy olla tavoitteellista ja se tulee kohdistaa sellaisille alueille, joille innovaatioita kaivataan. Kohdistettujen ideoiden arviointi ja seuranta on helpompaa. Lisäksi haettavien innovaatioiden tyyppi täytyy olla selvillä ennen ideoinnin aloittamista. Jos halutaan pysyä pienten parantelujen linjalla nykyisiin tuotteisiin, vaatii se huomattavasti erilaista ideointia kuin esimerkiksi voimakkaampaa kasvua tavoittelevat liiketoimintainnovaatiot. Myös asiakkaat ja heidän tarpeensa sekä yrityksen ydinosaaminen ja mahdollisuudet hankkia uutta osaamista tarvittaessa ohjaavat ideoinnin suuntaa. (Apilo ym. 2007, 139.)

Monissa ideointitekniikoissa pyritään mahdollisimman suureen ideoiden määrään. Näin saadaan käsiteltyä ideat, jotka tulevat ensimmäiseksi mieleen. Vasta näiden ideoiden käsittelyn jälkeen radikaalit uudet ideat pääsevät esille. Vaikka radikaalit ideat ovat tärkeitä, niiden täytyy olla tasapainossa tarkasti suunnattujen ideoiden kanssa yrityksen innovaatiostrategian mukaisesti. (Apilo ym. 2007, 140 - 141.)

Yleisin ideointiin käytetty tekniikka on yrityksen sisäinen aivoriihi (brainstorming). Aivoriihessä vaihteleva määrä henkilöitä kokoontuu yhteen esittämään ideoita, ajatuksia ja tietoa. Tärkeintä aivoriihen toiminnan kannalta on, että kaikki osallistujat pääsevät esittämään ideoitaan. Aivoriihen sujuvuuden kannalta on tärkeää, että kritiikkiä toisten ideoista ei esitetä ideoinnin aikana. Aivoriihen loppuksi esitetyt ideat käydään läpi. Tässä vaiheessa esitettyjä ideoita voidaan parannella ja yhdistellä toisiinsa. Ideoita voidaan parannella esimerkiksi siten, että jokaisesta ideasta pyritään löytämään ainakin kolme hyvää puolta ja tämän jälkeen kritiikki muotoillaan kehitysidean muotoon. (Apilo ym. 2007, 141; eNorssi, 2007.)

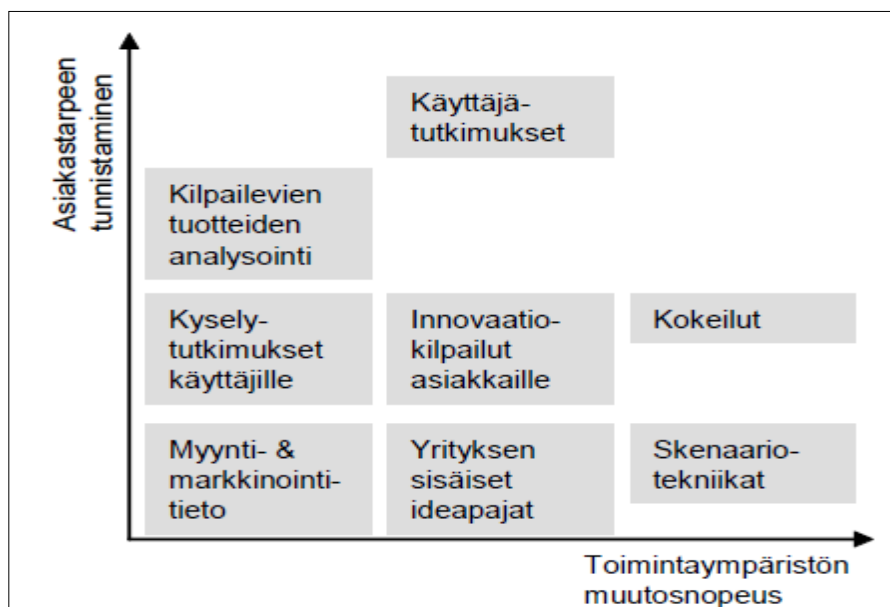
Ideointiin kannattaa käyttää mahdollisimman suurta joukkoa eri tasoilla toimivia henkilöitä. Muutaman suunnittelijan tai johtajan ideoidessa saadaan luultavasti aikaan huonompia ideoita, kuin ottamalla ideointiin mukaan henkilöitä yrityksen jokaiselta tasolta, kuten esimerkiksi huollosta. Lisäksi esimerkiksi toimittajaverkkoa ja asiakkaita kannattaa hyödyntää ideoinnin apuna. On myös mahdollista käyttää yrityksen sisäistä tai ulkopuolista konsulttia ideoinnin tehostajana. (Apilo ym. 2007, 141.)



Asiakkaita kannattaa käyttää ideoinnin apuna, jos se on mahdollista. Tätä puoltavat useat tutkimukset, joista käy ilmi, että asiakkaan kanssa kehitetyt innovaatiot ovat menestyneet hyvin. Asiakkaan kanssa ideoidessa kannattaa muistaa, että asiakas ei luultavasti osaa sanoa suoraan millaisen tuotteen hän haluaa. Asiakas osaa kuitenkin mahdollisesti kertoa millaisiin ongelmiin hän haluaa ratkaisun. (Apilo ym. 2007, 142.)

Käytettävä ideointitapa riippuu myös toimintaympäristöstä johon innovaatiota ollaan kehittämässä. Hitaasti muuttuvilla perinteisillä aloilla, joissa tuotteiden elinkaaret ovat pitkiä, kilpailijoiden tuotteiden analysointi, myynti- ja markkinointitiedon kerääminen sekä markkinatutkimukset ovat toimivia tapoja. Nopeammin muuttuvassa ympäristössä toimiville käyttäjätutkimukset, asiakkaille osoitetut ideakilpailut ja yrityksen sisäiset ideapajat ovat parempi vaihtoehto. Erittäin nopeasti muuttuvilla aloilla, kuten esimerkiksi mobiiliteknologia, toimivimmat tavat ideointiin ovat kokeilut ja uusien skenaarioiden luominen. (Apilo ym. 2007, 142.)

Kuvassa 3 on esitelty ideointitapoja toimintaympäristön muutosnopeuden mukaan. Näin tarkkaa jaottelua ei todellisessa tilanteessa kannata kuitenkaan noudattaa, vaan käytettävät ideointitavat kannattaa valita tilanteen ja yrityksen toimintatapojen mukaan.



KUVA 3. Ideointikeinoja erilaisiin toimintaympäristöihin. Jokaisen yrityksen täytyy valita itselleen sopivimmat keinot toimintaympäristönsä perusteella. (Apilo & Taskinen 2006, 50.)

Ideoiden määrä ei yleensä ole ongelma. Ongelmaksi voi muodostua ideoiden kerääminen ja ideoiden tyyppin määrittäminen. Tärkeää ideoiden keräämisen edistämiseksi on, että ideoiden esittäminen on tarpeeksi helppoa koko henkilöstölle. Ideoiden esittämisen foorumina voidaan käyttää esimerkiksi yrityksen intranettia, jonne kaikilla on pääsy. Lisäksi henkilöstöä tulisi kannustaa esittelemään ideansa muille ja esiteltyjä ideoita varten tulisi luoda nopeasti toimiva arviointisysteemi. Jos ideaa ryhdytään viemään eteenpäin, on tärkeää, että idean kehittäjä saa tästä nopeasti tietoa. Myös henkilöiden, jotka osallistuvat konseptointiin (katso 3.3.3), täytyy saada ideat nopeasti käyttöönsä. (Apilo ym. 2007, 143 - 144.)

Kohdennettuja ideoita voidaan jatkuvan keräämisen lisäksi saada esille järjestämällä innovaatiokilpailuja. Innovaatiokilpailuja voidaan järjestää yrityksessä sisäisesti, asiakkaille, muille sidosryhmille sekä myös opiskelijoille. Opiskelijoille järjestetyt innovaatiokilpailut luovat uusia ideoita ja saavat opiskelijat kiinnostumaan yrityksestä. (Apilo ym. 2007, 144 - 145.)

Hylätyistä ideoista tulisi ylläpitää tietokantaa. Näitä ideoita voidaan mahdollisesti hyödyntää myöhemmässä vaiheessa, riippuen siitä, miksi idea on aikaisemmin hylätty. Esimerkiksi teknologian kehittyminen voi mahdollistaa jonkin aikaisemmin mahdottomalta tai liian haasteelliselta tuntuneen idean kehittämisen. (Apilo ym. 2007, 146.)

### 3.3.1.3 Ideoiden kehittäminen

Ideointivaiheessa keksittyjä ideoita täytyy jalostaa eteenpäin kypsemmiksi ideoiksi ennen prosessin seuraavaa vaihetta. Jos ideoita ei kehitetä yhdessä ennen arviointivaihetta, voivat hyvätkin ideat tulla hylätyiksi. Ideoiden kehittäminen on innovaatioprosessin alkupään haasteellisimpia vaiheita. Innovatiiviset organisaatiot osaavat hyödyntää tämän vaiheen tarjoamat mahdollisuudet kunnolla. (Apilo ym. 2007, 148.)

Kehittämisen ensimmäisessä vaiheessa ideat asetetaan ideoijan tai ideointiryhmän ulkopuolisten henkilöiden jatkojalostettavaksi. Jatkojalostamiseen voidaan käyttää erilaisia työpajoja tai mahdollisille ostajille ja käyttäjille pidettäviä teknologiaseminaareja. Jatkokehittäminen voidaan toteuttaa myös virtuaalisessa ympäristössä. (Apilo ym. 2007, 148.)

Jatkokehityksen jälkeen kypsiä ideoita tulisi tarkastella kriittisesti yrityksen strategian, asiakastarpeen, käytettävien resurssien ja toteutusmahdollisuuksien kannalta. Tarkastelun avuksi voidaan toteuttaa esimerkiksi pienimuotoisia teknologiaselvityksiä, kokeiluprotot ja visualisointia. Edellä mainitut vaiheet toteutetaan, koska ideoiden arviointivaiheessa (katso 3.3.1.4) on vielä mahdollista esittää uusia ratkaisuja ja vertailla kilpailevia ehdotuksia keskenään. Kehitysvaiheessa vaikutusmahdollisuudet lopputulokseen yleensä vähenevät, vaikka muiden kiinnostus tuotteeseen lisääntyykin samalla kun tuote tulee konkreettisemmaksi. Myös johdon sitouttaminen tuotteeseen mahdollisimman aikaisessa vaiheessa auttaa siihen, että toteutettava innovaatio saadaan sovitetuksi liiketoimintaan. Johdon sitoutuminen voi helpottaa resurssien saamista projektille sen toteuttamisen aikana. (Apilo ym. 2007, 148.)

#### 3.3.1.4 Ideoiden arvioiminen

Ideoiden arvioinnissa päätetään idean toteutuksesta ja toteutuksen vaatimuksista. On myös mahdollista, että idea päätetään jättää odottamaan sopivampaa ajankohtaa. Idean toteutukseen vaikuttaa se, minkätyyppisestä ideasta on kyse. Tässä vaiheessa päätetään lisäksi, liitetäänkö idea osaksi käynnissä olevaa kehitys- tai tutkimushanketta vai käynnistetäänkö idean pohjalta uusi konseptointivaihe. (Apilo ym. 2007, 150.)

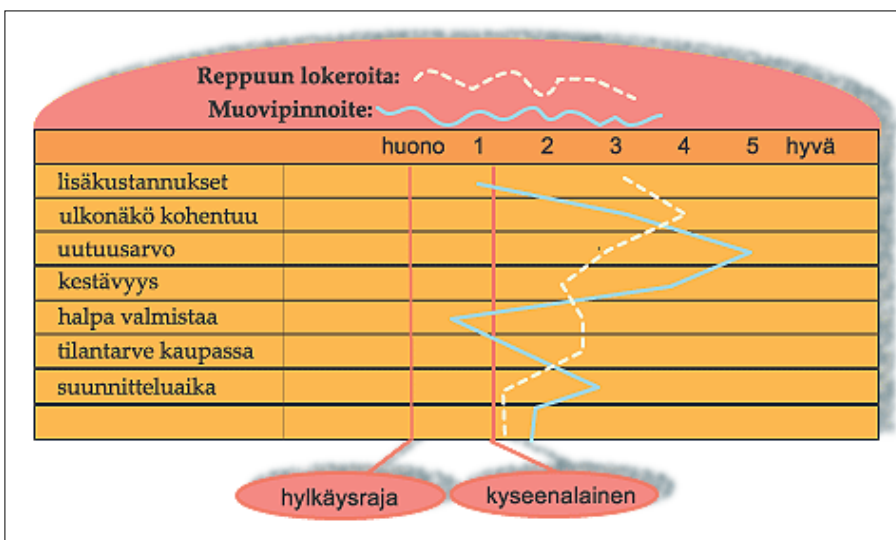
Idean toteutusmahdollisuudet ja idean sopivuus yritykselle sekä asiakastarpeet ovat pääsuunnat idean arvioimiselle. Idean täytyy olla teknisesti ja lainsäädännöllisesti toteutettavissa sekä sille täytyy olla tunnistettu asiakastarve. Lisäksi idean täytyy sopia yrityksen omaan tai verkostokumppaneiden visioon, resursseihin ja strategiaan. (Apilo ym. 2007, 151.)

Idean arvioinnin tukena voidaan käyttää esimerkiksi kuvissa 4 ja 5 esitettyjä menetelmiä. Esimerkkinä on käytetty uusien ominaisuuksien lisäämistä koulureppuun. Taulukkomenetelmässä (kuva 4) arviointikriteerin kohdalle merkitään sitä enemmän plusia, mitä parempi tarkasteltava idea kyseisen kriteerin kannalta tarkasteltuna on. Jos idea kriteerin kannalta tarkasteltuna on huono, merkitään taulukkoon kyseiseen kohtaan miinus. Samalla on mahdollista miettiä, miten kyseistä ideaa voitaisiin muuttaa niin, että se olisi toteuttamiskelpoinen myös tarkasteltavasta näkökulmasta. Ideat saadaan paremmuusjärjestykseen laskemalla lopuksi idean saamat plussat ja miinukset yhteen. Arvioinnissa voidaan käyttää myös numeerisia arvoja plussien ja miinusten sijasta. (Lavonen & Meisalo.)

	lisäkustannukset	ulkonäkö kohentuu	uutuusarvo	kestävyys	halpa valmistaa	tilantarve kaupassa	suunnittelu-aika
reppuun lokeroita	+	+++	+++	+			
muovipinnoite	+		++				
metallivahvikkeet kulmiin	++	++	+++				
kokoontaittava reppu	++		++++			+	
sivutaskuun kampa	-		++	++	+		
takaosaan heijastin	-		+				

KUVA 4. Repun kehittämiseen liittyvien ideoiden tarkastelu taulukon avulla. Arvioinnin lopuksi plussat ja miinukset lasketaan yhteen. Arvioinnissa voidaan käyttää myös numeerisia arvoja. (Lavonen & Meisalo.)

Graafisen esityksen avulla voi olla helpompi hahmottaa, mitkä ideat ovat toteuttamiskelpoisia ja mitkä kannattaa hylätä. Spektrimenetelmässä (kuva 5) jokaisesta ideasta piirretään kuvaaja, jonka pisteet asetetaan sen mukaan, miten hyvin idea kunkin kriteerin täyttää. Jos jokin kuvaajan piste ulottuu kyseenalaiselle alueelle, on se tämän kriteerin kannalta toteuttamiskelvoton.



KUVA 5. Repun kehittämiseen liittyvien ideoiden tarkastelu spektri-arviointin avulla. Jokaisesta ideasta piirretään kuvaaja. Kyseenalaiselle alueelle joutuva idea on kyseisen kriteerin kannalta toteuttamiskelvoton. (Lavonen & Meisalo.)

### 3.3.2 Palkitsemiskäytännöt

Oikein toteutetulla palkitsemiskäytännöllä voidaan vaikuttaa huomattavasti ideoiden esilletuloon sekä niiden kehittämiseen ja hyödyntämiseen yrityksen tuotteissa. Palkitsemiskäytäntöjen perinteinen muoto, rojaltipohjainen palkitseminen, suosii yksin suoritettua kehitystä ja patentoitavaa teknologista keksintöä. Tämä ei yrityksen kannalta ole tuloksellisin ratkaisu. Rojaltipohjainen palkitseminen kannustaa kehittämään ideaa yksin ja suojelemaan ratkaisua patentoinnin jälkeen. Pahimmassa tapauksessa tämä estää ratkaisun jatkokehityksen. Eräs vaihtoehto, joka voi helpottaa keksintöjen esilletuontia ja hyödyntämistä, on siirtyä rojaltipohjaisesta palkkiosta kertakorvauksiin. (Apilo ym. 2007, 148.)

### 3.3.3 Konseptointi

Toteutusvaihe alkaa konseptoinnilla ja esisuunnittelulla. Yleensä konseptointi ja esisuunnittelu kannattaa yhdistää. Aloituksen jälkeen toteutusvaiheessa siirrytään kehitys- ja testausvaiheeseen. Kehitys- ja testausvaiheet voivat toistua useita kertoja prosessin aikana. (Apilo ym. 2007, 131.)

Tässä luvussa käsitellään konseptointia yleensä sekä konseptoinnin eri tasoja tarkemmin. Esisuunnittelua ei käsitellä erillisenä prosessin osana.

#### 3.3.3.1 Konseptoinnin määritelmä

*Tuotekonseptilla tarkoitetaan yleensä tuotehahmotelmaa, jolla ei suoraan ohjeisteta tuotantoa eikä mennä markkinoille, vaan sen avulla tutkitaan mahdollisuuksia ja tehdään päätöksiä. Tuotekonseptiin sisältyy normaalisti informaatio tuotteen toiminnasta, kohderyhmästä, käytetystä teknologiasta, rakenteesta yms. Osa tuotekonsepteista jalostetaan tuotantovalmiiksi asti, osa voi olla tarkoitettu yrityksen muuten hyödynnettäviksi tutkielmiksi.* (Kokkonen ym. 2005, 11.)

Konseptoinnin hahmottamisessa ja toteutuksessa on useita haasteita. Konseptointi vie aikaa ja resursseja ennen tuotekehityksen alkamista. Konseptoinnin aikana tehdyt ratkaisut ovat kuitenkin merkittäviä asiakastarpeen toteutumisen ja toteutuksen hinnan kannalta. Jos perusteellisen konseptoinnin lisäksi vaaditaan nopeaa tuotekehitystä, voi tavoitteiden välille syntyä ristiriitatilanteita. Ennen konseptoinnin aloitusta joudutaan myös miettimään konseptoinnin tyyliä. Täytyy päättää, valitaanko konseptointiin vapaa ja joustava vai tiukasti ohjattu lähestymistapa. Lisäksi teknologia- ja markkinalähtöisyys täytyy konseptoinnissa saada tasapainoon. (Apilo ym. 2007, 151.)

Tämän luvun alussa on esitetty eräs tuotekonseptin määritelmä. Konseptoinnin hahmottamista voi kuitenkin vaikeuttaa sen monikäsitteisyys. Konseptoinnilla voidaan tarkoittaa aikaisemman innovaatioprosessin alkupään aikana tapahtuvaa vapaata ajattelua, yleistä teknologisten ratkaisujen kehittelyä, projektoimattoman ja projektoidun innovaatioprosessin välissä olevaa erillistä vaihetta, tuoteprosessin ensimmäistä vaihetta tai kaikkia edellä mainittuja (Apilo ym. 2007, 151).

### 3.3.3.2 Konseptoinnin kategoriat

Konseptointi voidaan jakaa eri kategorioihin. Apilo, Taskinen ja Salkari (2007, 152) esittävät Perttulan ja Sääksilahden (2004) kehittämät neljä konseptoinnin kategoriaa: visioiva (visioning), kehittävä (emerging), määrittelevä (defining) ja ratkaiseva (solving) konseptointi. Näistä visioiva ja kehittävä konseptointi ovat pidemmälle tulevaisuuteen tähtäviä ja ne eivät välttämättä suoraan johda kaupalliseen tuotteeseen. Määrittelevä ja ratkaiseva konseptointi taas liittyvät suoraan tuotekehitysprojektiin. (Kokkonen ym. 2005, 17.)

Taulukossa 1 on käyty läpi konseptointikategorioiden ominaispiirteitä. Konseptointitasot on esitetty niin, että alimpana taulukossa on lähimpänä toteutusta toimiva konseptointitaso. Seuraavissa luvuissa nämä konseptointitasot on esitelty tarkemmin. Koska asiakastarpeiden määrittäminen eri vaiheissa on tärkeää lopullisen ratkaisun kannalta, on siihen liittyviä asioita käsitelty jokaisen konseptointitason kohdalla tarkemmin.

TAULUKKO 1. Eri konseptointikategorioiden ominaispiirteitä (Apilo ym. 2007, 152.)

Konseptointitaso	Tavoite	Toteutustapa	Haasteita	Mahdollisuuksia
Visioiva konseptointi	Konkretisoida tulevaisuuden mahdollisuuksia	Jatkuva strategisen ajattelun prosessi	Ajan ja oikeiden henkilöiden löytäminen, myöhemmän ideoinnin rajoittaminen	Konkretisoi visioita, helpottaa päätöksen tekoa, auttaa luomaan mielikuvaa edelläkävijästä (sijoittajat, rekrytointi)
Kehittävä konseptointi	Konseptimoduuleita toteutus suunnitteluun	Jatkuva prosessi	Ajan löytäminen	Lyhentää toteutusvaihetta, luo hyviä ja vanhaa kyseenalaistavia konsepteja
Määrittelevä konseptointi	Ominaisuus- ja ratkaisumäärittelyyn vaihtoehtoisia konsepteja	Esisuunnittelu projekteissa	Poikkifunktionaalisten resurssien hyödyntäminen	Konkretisoi vaihtoehtoja ja parantaa päätöksentekoa
Ratkaiseva konseptointi	Pohja määrittelyvaiheelle	Kehitysprojekteissa	Malttia kehittää useampia vaihtoehtoisia konsepteja	Luo toteutukseen osallistujia innostavia ja asiakastarvetta hyvin vastavia ratkaisuja

### 3.3.3.3 Visioiva konseptointi

Visioivan konseptoinnin avulla pyritään seuraamaan tulevaisuuden tarjoamia mahdollisuuksia ja se auttaa tunnistamaan sekä yllättäen eteen tulevia radikaaleja innovaatioita että pidemmän aikavälin innovaatioita. Visioivaa konseptointia voidaan käyttää niin tuotteen, palvelun kuin liiketoiminnan strategian konkretisoinnissa ja se toimii strategisen suunnittelun työvälineenä. Lisäksi visioivan konseptoinnin avulla voidaan selvittää, mihin yrityksen tutkimusresurssit kannattaa suunnata. Visualisointi on tärkeä visioivan konseptoinnin osa-alue. Tässä vaiheessa visualisoinnin avulla idealle annetaan ensimmäinen konkreettinen muoto joka helpottaa päätöksentekoa ja sitoutumista sekä auttaa luomaan mielikuvia. (Apilo ym. 2007, 153–154.)

Asiakastarpeen määrittäminen on haasteellista visioivan konseptoinnin aikana, koska mahdolliset asiakkaat eivät ole välttämättä selvillä konseptoinnin aluksi. Asiakastarpeen määrittäminen tulee aloittaa asiakkaiden määrittämisellä ja valitsemisella. Kun mahdolliset asiakkaat on saatu valittua, siirrytään ennakoimaan asiakkaiden tulevaisuuden tarpeita ja tarpeiden kehittymistä. Lopullisena tavoitteena visioivan konseptoinnin aikana on pyrkiä ymmärtämään valittua asiakasta tai asiakkaan liiketoimintaprosessia, jos kyseessä on yritysasiakas. (Apilo ym. 2007, 158.)

### 3.3.3.4 Kehittävä konseptointi

Kehittävän konseptoinnin tärkeimpiä ohjaavia tekijöitä ovat asiakkaan tarpeet. Kehittävää konseptointia voidaan toteuttaa yhdessä asiakkaiden kanssa. Asiakkailta kerättyä tietoa pyritään hyödyntämään konseptoinnin aikana jatkuvasti. Asiakastarpeiden lisäksi kehittävän konseptoinnin avulla voidaan tunnistaa ja määritellä tarpeita yrityksen teknologian kehitykselle etsimällä teknologisia ideoita. (Apilo ym. 2007, 154–158.)

Kehittävän konseptoinnin tulisi olla jatkuva prosessi jossa on koko ajan kehitteillä eriasteisia konsepteja. Konseptoinnin aikana arvioidaan milloin konsepti on niin kypsä, että se voidaan liittää osaksi sopivaa toteutusprojektia. Esimerkkinä voidaan käyttää videopelin käyttöliittymän kehittämistä. Kun konseptoinnin arvioinnissa havaitaan, että käyttöliittymä on tarpeeksi kypsä, se liitetään osaksi toteutusprojektia. (Apilo ym. 2007, 154–158.)

### 3.3.3.5 Määrittelevä konseptointi

Määrittelevän konseptoinnin aikana mietitään mahdolliset tuotteen elinkaaren aikana käytettävät liiketoiminta- ja ansaintamallit. Tuoteinnovaation kohdalla tuoteratkaisun rinnalla voidaan konseptoida tuotteeseen liittyviä palveluita ja ansaintalogiikoita. Esimerkiksi ansaintalogiikkaa koskevassa rinnakkaiskonseptissa voidaan päättää, koostuvatko ansiot tuotteiden myynnistä vai varaosista. Määrittelevän konseptoinnin aikana tuotetut konseptit voivat myös toimia karkean tason määrittelyinä kehitysprojekteille. (Apilo ym. 2007, 155.)

Määrittelevässä konseptointivaiheessa asiakastarpeet pyritään täyttämään ongelmanratkaisukeskeisesti. Tässä vaiheessa asiakastarpeiden määrittelyn ja täyttämisen kannalta haasteellista on asiaankuuluvien tarpeiden valitseminen ja valittujen tarpeiden muuntaminen tuotteen ominaisuuksiksi. (Apilo ym. 2007, 159.)



### 3.3.3.6 Ratkaiseva konseptointi

Ratkaisevan konseptoinnin aikana yhdistetään tunnistettuja asiakastarpeita sekä olemassa olevia ratkaisuja ja tekniikoita liiketoimintanäkemyksen mukaiseksi kokonaisuudeksi. Tuotekehitysprojekti voidaan aloittaa jo ratkaisevan konseptoinnin aikana. Tuotekehitysprojektin aikana ratkaisevan konseptoinnin avulla tutkitaan, kuinka hyvin valitut ratkaisut vastaavat asiakastarpeita. Tuotteeseen voidaan myös lisätä kehitysprojektin aikana havaittuja tarpeita. (Apilo ym. 2007, 155–159.)

Poikkifunktionaalisten konseptointitiimien käyttäminen on erityisen tärkeää ratkaisevan konseptoinnin aikana. Johto- ja tuotekehitysryhmän lisäksi tiimeihin tulisi kerätä mahdollisimman paljon henkilöitä esimerkiksi seuraavista ryhmistä: myynti, markkinointi, tuotanto, logistiikka, huolto ja varaosatoimitukset. Näin pystytään varmistamaan konseptin hyvä laatu. (Apilo ym. 2007, 156.)

### 3.3.3.7 Yhteenveto konseptoinnista

Lopullisen ratkaisun kannalta konseptoinnilla on suuri merkitys, ja siksi poikkifunktionaalisen osaamisen käyttäminen konseptoinnin aikana on tärkeää. Sopivimmat konseptointitavat yritykselle löytyvät kokeilemalla. Sopivien konseptointitapojen lisäksi on tärkeää löytää ne henkilöt, joilla on tarvittavia kykyjä ja tietämystä sekä halua osallistua konseptointiin. (Apilo ym. 2007, 156.)

Konseptoinnin aikana saatava palaute on tärkeää tuotekehitysprojektin onnistumisen kannalta. Paras tapa saada palautetta asiakkailta on ottaa asiakkaat konseptointiin mukaan. Näin tekemällä konseptit saadaan ohjautumaan asiakastarpeiden ja asiakkaiden toimintatapojen mukaisesti. Konseptin esittelyä helpottaa sen visualisointi ja erilaiset prototyypit. Selkeästi esitetyllä konseptilla on mahdollista saada hyödyllistä, projektin toteutukseen vaikuttavaa palautetta. (Apilo ym. 2007, 156–159.)

Ennen kehitetyn konseptin siirtämistä eteenpäin se täytyy arvioida. Arviointiin käytettävät kriteerit tulee valita konseptin tyyppin ja yrityksen tarpeiden mukaisesti. Kriteereihin voi kuulua konseptin toiminnan ja ominaisuuksien arvioinnin lisäksi esimerkiksi käytettävän tekniikan ja konseptin markkinapotentiaalin arviointi. Tärkeintä konseptin arvioinnissa on, että se suoritetaan tarpeeksi suurella foorumilla. Alkavien hankkeiden priorisoinnin helpottamiseksi eri konsepteille kannattaa pyrkiä käyttämään samankaltaisia arviointikriteereitä. (Apilo ym. 2007, 158.)

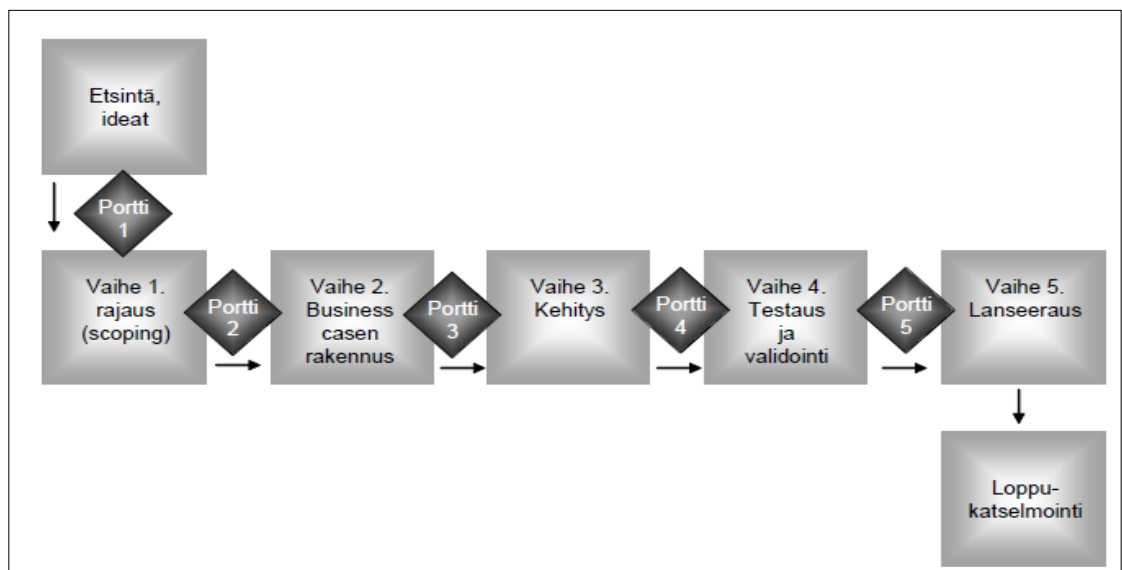
### 3.3.4 Tuoteprosessi

Fyysisen tuotteen kehittämisen yhteydessä tuoteprosessilla tarkoitetaan yleensä projektoitua ajanjaksoa, joka alkaa konseptoinnista ja päättyy tuotteen volyymivalmistuksen alkamiseen. Projektionin määrittelyssä yritysten välillä syntyy eroja pääasiassa siinä, miten konseptointi sijoitetaan osaksi tuotekehitysprojektia eli suoritetaanko konseptointi kokonaisuudessaan tai osittain ennen projektia vai vasta projektin aikana. Nopeus on yleensä tärkein yksittäinen tuoteprosessille määritelty ominaisuus. Nopeuteen voidaan päästä esimerkiksi hyödyntämällä verkoston yhtäaikaisen suunnittelun mahdollisuutta sekä käyttämällä poikkifunktionaalisia tiimejä. (Apilo & Taskinen 2006, 53.)

Yleisesti ottaen tuotekehitysprojektien tuoteprosessi on yrityksissä hyvin hallussa ja sitä toteutetaan määriteltyjen prosessikuvausten mukaisesti. Parantamista monella yrityksellä on poikkifunktionaalisen lähestymistavan tehokkaammassa hyödyntämisessä ja tämän lähestymistavan kirjaamisessa osaksi prosessia. Tuoteprosesseissa, joissa on ollut mukana esimerkiksi markkinointi, logistiikka sekä huolto, on prosessin läpivienti ollut vaivattomampaa ja projektin loppuvaiheessa tuotemuutoksia on tarvittu normaalia vähemmän. (Apilo & Taskinen 2006, 54.)

Tarkasti kuvatun prosessin hyöty on, ettei jokaisen projektin alussa tarvitse suunnitella uudelleen, kuinka projektissa tullaan toimimaan, vaan ideointi voidaan kohdistaa itse tuotteeseen. Lisäksi yritykseen tai projektiin mukaan tulevat uudet henkilöt saavat nopeammin kuvan yrityksen toimintatavoista. Huonoina puolina tarkasti kuvatuissa prosesseissa voidaan nähdä byrokratian lisääntyminen sekä joustamattomuus, jos muutostarpeita huomataan prosessin aikana. Myös tarvittavien dokumenttien viimeistelyyn voi mennä huomattavasti aikaa, mikä voi hidastaa prosessin etenemistä etenkin, jos dokumentteja tarvitaan prosessin seuraavissa vaiheissa. Rinnakkaisten toimintojen mahdollistaminen on yksi tärkeimmistä tavoitteista tarkasti kuvatuissa prosesseissa. (Apilo & Taskinen 2006, 54.)

Yksi mahdollisuus tuoteprosessin hallintaan on Apilon ja Taskisen (2006) esittämä Cooperin (1993, 1999) porttimalli, joka on käytössä monella Suomalaisella yrityksellä. Porttimalli täytyy räätälöidä yrityskohtaisesti ja sen suunnitteluun tulee ottaa mukaan koko sidosryhmä. Porttimallin avulla prosessiin saadaan selkeät tarkastuspisteet. Tarkastuspisteissä katselmoidaan projektin eteneminen ja käsitellään esille nousseet asiat sekä tuotetaan tilanneraportit johdolle. Mallin mukaisesti tarkastuspisteissä tulee arvioida, ollaanko valmiita siirtymään seuraavaan vaiheeseen vai täytyykö edelliseen vaiheeseen vielä palata. Lisäksi tulee arvioida, kannattaako projektia jatkaa eteenpäin vai keskeyttää projekti. Projekti kannattaa keskeyttää, jos esimerkiksi markkinatilanne tai asiakasvaatimukset ovat selkeästi muuttuneet. Keskeyttämällä projekti siihen varatut resurssit on mahdollista sijoittaa muuttuneeseen tilanteeseen paremmin sopiviin projekteihin. Yleensä tarkastuspisteitä ei kuitenkaan täysin osata hyödyntää, vaan projekti pyritään viemään loppuun muuttuneesta tilanteesta huolimatta. Porttimallin esimerkki on esitetty kuvassa 6. (Apilo & Taskinen 2006, 57 - 58.)



KUVA 6. Cooperin porttimalli. Eri vaiheiden välisiä portteja käytetään tarkastuspisteinä. Näissä pisteissä tulee arvioida, ollaanko valmiita siirtymään seuraavaan vaiheeseen. Lisäksi tulisi arvioida kannattaako projektia jatkaa eteenpäin vai keskeyttää projekti. (Apilo & Taskinen 2006, 59.)

Ohjelmistoprojektin toteutus eroaa jonkin verran perinteisestä, pelkkää mekaniikkaa sisältävästä tuotekehitysprojektista (Apilo & Taskinen 2006, 58). Ohjelmistoprojektin erikoispiirteitä on esitelty luvussa 5.2.

### 3.3.5 Innovaation kaupallistaminen ja markkinointi

Aiemmin esitettyyn innovaation määritelmään kuului, että sen tavoitteena on taloudellisen hyödyn saaminen. Innovaation kaupallistamiseen on Fogelholmin (2009, 15) mukaan käytännössä kaksi vaihtoehtoa:

1. Tuoteoikeuksien myynti tai lisensioiminen ulkoiselle yritykselle
2. Tuotteen kaupallistaminen (valmistaminen ja/tai myynti) omana liiketoimintana

Lisensiointi voi olla haasteellista, jos tarjottava tuote ei ole vielä täysin valmis tai sille ei ole olemassa varmoja markkinoita. Uusien markkinoiden luominen voi lisenssinottajalle olla ylivoimaista. Lisenssin tarjoaja joutuu miettimään, onko lisenssinottajalla tarvittavat resurssit tuotteen kaupallistamiseen, koska rojalTIMaksut on mahdollista saada vain myydyistä tuotteista. On myös olemassa vaara, että lisenssinottaja kehittää ajan kuluessa oman, kilpailevan tuotteen ja lopettaa rojalTIMaksut kokonaan. Lisensioinnin aiheuttamien riskien todennäköisyyttä voidaan vähentää sillä, että molemmat osapuolet tiedostavat ne lisensiointia aloitettaessa. (Fogelholm 2009, 46; Korkiakangas 1998, 33.)

Lisensioinnin etuja sekä lisenssin tarjoajalle että ottajalle on useita. Korkiakangas (1998, 33) luettelee esimerkiksi seuraavat edut: säästöjen saavuttaminen tuotekustannuksissa ja tuotekehityksen nopeuttaminen, teknologian kehityksessä mukana pysyminen, uusille markkinoille pääseminen ja vanhojen laajentaminen sekä aukkojen täyttäminen tuotevalikoimassa.

Käytännössä tuotteiden oma valmistus ja myynti jäävät usein keksijän ainoaksi tavaksi hyötyä keksinnöstään. Tämä johtuu siitä, että tuotteen lisensiointi tai tuoteoikeuksien myynti kertamaksulla onnistuvat vain harvoin. Keksintöjen piirissä tämä yleensä vaatii yrityksen perustamisen tuotetta varten. Jos tuoteinnovaatio on syntynyt jo olemassa olevan yrityksen sisällä, on se mahdollista ottaa yrityksen valikoimaan mukaan. (Fogelholm 2009, 58.)

On myös olemassa teknologian aloja, joilla täydelliset markkinointitaidot eivät ole välttämättömiä kaupallisen menestymisen kannalta. Terveysteknologiassa, jossa alan suomalaiset yritykset ovat pärjänneet hyvin, tuotteen tekninen laatu ratkaisee pitkälti sen kaupallisen menestyksen. (Ojanperä 2012, 5.)

Optomedin Seppo Kopsala kuvaa markkinoinnin roolia terveysteknologia-alalla Ojanperän (2012, 5) mukaan seuraavasti:

*Lääketieteellisissä laitteissa tuotteiden tekninen laatu ja asiakkaan saama todellinen lisäarvo vaikuttavat enemmän kuin markkinoinnin kautta luotu tuoteimago ja halvin mahdollinen hinta. Siksi tämä ala sopii hyvin suomalaiseen tapaan toimia.*

Tuotteen kaupallistamisen ja markkinoinnin syvällisempi käsittely on rajattu tämän työn ulkopuolelle. Luvussa 4 on käsitelty mobiilisovellusten erilaisia jakelukanavia sekä sovellusten markkinointiin liittyviä asioita yleisellä tasolla.

## 4 MOBIILISOVELLUSTEN MARKKINOINTI JA JAKELU

Markkinoinnilla on tärkeä osa menestyksekkäässä innovaatioprosessissa, myös kehitettäessä uusia mobiilisovelluksia, koska luvussa 3.1 esitetyssä innovaation määritelmässä taloudellinen menestyminen on yksi osa innovaatiota. Seuraavissa luvuissa on käsitelty mobiilisovellusten markkinointia yleisellä tasolla esimerkkien avulla. Lisäksi on esitelty Applen, Microsoftin ja Googlen tarjoamat mobiilisovellusten jakelukanavat.

### 4.1 Yleistä mobiilisovellusten markkinoinnista

Olli Sinermä (2012, 62) kuvaa itsenäisten pelintekijöiden ongelmia erottua massasta. Suurin ongelma erottumisen kannalta on se, että pelintekotyökalut ovat ilmaisia ja kuka tahansa voi tehdä esimerkiksi Applen Appstoren myyntilistojen kärkeen pääsevän pelin jopa ilman ohjelmointitaitoja. Lisäksi on todennäköistä, että pelin saatua jonkin verran suosiota, siitä tullaan tekemään samankaltaisia kilpailevia ja ilmaisia versioita. Markkinoinnin vaikutuksesta hän kirjoittaa seuraavaa:

*Yksinäisen indiedevaajan pitää ilman lisenssilihasta joko luottaa sokeasti viraaliin ja kaverien tykkäyksiin Facebookissa, tai tunkea muutama kymppitonni mainontaan, jotta peli saa edes vähäisen näkyvyyden kymmenien tuhansien samalta linjalta spurttaavien kilpailijoiden joukosta.*

Mobiilisovellusten maailmassa markkinointiin ei aina tarvita erityisen suurta budjettia. Esimerkkinä voidaan käyttää suomalaisen Remedy-pelityhtiön Death Rally -nimisen pelin mobiiliversion julkaisua ja markkinointia, joista Tero Kuittinen (2012) kirjoittaa Forbesin nettikolumnissaan. Remedy onnistui tekemään miljoonan dollarin myynnin 10 000 dollarin markkinointibudjetilla. Tässä onnistuttiin kohdistamalla markkinointibudjetti järkevästi.

Kun Death Rally julkaistiin App Storessa maaliskuussa 2011, Remedy asetti pelille suhteellisen korkean hinnan, 4,99 dollaria, verrattuna toisiin peleihin. Markkinointiin käytettiin edellisessä kappaleessa mainittu 10 000 dollaria: 5000 dollaria FreeAppA-Day-kamppanjaan ja 5000 dollaria tarkasti kohdistettuun PR-työhön. FreeAppADay-kampanjassa mobiilisovellusten leviämistä edistetään jakamalla niitä ilmaiseksi rajoitettu aika. Huolimatta korkeasta hinnasta Remedy sai kehityskustannuksensa takaisin kolmessa päivässä julkaisun jälkeen. Toukokuussa 2011 Remedy pudotti pelin hintaa 0,99 dollariin. Tämä aiheutti sen, että pelistä tuli suosituin iPhone-sovellus maailmassa viiden päivän ajaksi. Voidaan ajatella, että pelin alkuperäinen hinta antoi sille laadukkaan tuotteen leiman. Kun pelin hintaa laskettiin, hyvien arvioiden vuoksi pelistä kiinnostuneet henkilöt innostuivat hankkimaan sen. Kirjoitushetkellä peli oli ohittanut 10 miljoonan latauksen rajan. (Kuittinen 2012; Nirvi 2012.)

Edellä kuvatut asiat voidaan yleistää myös muihin loppukäyttäjille tarkoitettuihin, yleisesti jakelussa oleviin mobiilisovelluksiin. Jos kyseessä on valmiiksi myyty B2B-sovellus, ei enää toteutusvaiheen jälkeen tarvita mittavaa markkinointia. Luvussa 3.3.5 mainittiin, että keksijä joutuu monesti perustamaan uuden yrityksen, jos haluaa hyötyä keksinnöstään. Mobiilisovellusten valmistus, markkinointi ja jakelu eivät välttämättä vaadi niin suuria panostuksia, että uuden yrityksen perustaminen olisi ainoa vaihtoehto menestymisen kannalta. Itsenäinen sovelluskehittäjä, kuten yritysikin, voi laittaa sovelluksen myyntiin jakelijan verkkokauppaan, ja se voi menestyä hyvin. Jos markkinointiin halutaan panostaa enemmän, voi kehittäjä lisensoida tuotteensa jollekin yritykselle tai hankkia rahoitusta esimerkiksi sijoittajilta.

## 4.2 Apple

Sovellusten jakelu Applen mobiililaitteille suoritetaan pääasiassa App Store -nimisen sovelluskaupan kautta. Sovelluksen kehittäjä toimittaa sovelluksen myyntiin ja asettaa sille mieleisensä hinnan. Myyntien sovellusten hinnasta Apple ottaa 30 % itselleen ja tilittää loput sovelluksen kehittäjälle. Ilmaisten sovellusten jakelusta ei peritä maksua. Myyntihinnan lisäksi Apple tarjoaa kolmea erilaista tapaa ansaita sovelluksilla: sovelluksen sisäiset ostot, mainokset ja volyymiostot. Sovelluksen sisäiset ostot voivat olla esimerkiksi uusia kenttiä tai lisäominaisuuksia johonkin peliin. Mainoksia esitetään Applen iAd Network -palvelun kautta. Mainostuloista sovelluksen tekijä saa 70 %. (Apple 2012.)

Volyymiostot on tarkoitettu pääasiassa yrityksille tai oppilaitoksille. Volyymiosto-ohjelmaan liittyneet asiakkaat voivat ostaa suuria määriä sovelluksen käyttöoikeuksia kerralla. Volyymiosto-ohjelmaan liittyneille asiakkaille on mahdollista myydä myös B2B-sovelluksia. Tällöin sovellusta ei tarvitse laittaa yleiseen jakeluun App Storen kautta. (Apple 2012.)

### 4.3 Microsoft

Windows Phone -sovellusten jakelu hoidetaan Marketplace-nimisen kauppapaikan kautta. Valmiit sovellukset toimitetaan Microsoftille hyväksyttäväksi App Hub -nimisen palvelun kautta. App Hubiin on keskitetty kaikki sovellusten jakeluun ja sovelluskehityksessä tarvittavaan tukeen liittyvät asiat, kuten esimerkiksi kehittäjäyhteisön foorumit. Ennen kuin sovelluksen toimittaminen tarkastukseen on mahdollista, täytyy kehittäjän maksaa Microsoftille 99 dollarin rekisteröitymismaksu. Maksu täytyy uusia vuosittain. Lisäksi rekisteröitymiseen tarvitaan Windows Live ID. Rekisteröitymismaksun lisäksi Microsoft ottaa 30 % myytyjen sovellusten tuloista itselleen. (Microsoft 2012, Marketplace; Microsoft 2012, App Hub.)

Hyväksynnän jälkeen sovellukset siirtyvät automaattisesti Marketplaceen ladattavaksi. Tälle prosessille ei tällä hetkellä ole vaihtoehtoja, ja esimerkiksi sovellusten testaus kehitysvaiheessa on mahdollista vain tarkoitukseen rekisteröidyllä päätelaitteella. Beta-testausvaiheessa on mahdollista jakaa sovellus valituille henkilöille kokeiltavaksi ennen sen julkaisua yleiseen jakeluun. Beta-testauksessa sovellus toimitetaan käytettäväksi henkilöille, jotka eivät ole olleet sen kehityksessä mukana. Näin saadaan parempi käsitys sovelluksen toiminnasta sekä mahdollisista virheistä ja ongelmista käyttäjän näkökulmasta. App Hubista täytyy valita sovelluksen toimituksen yhteydessä, että se halutaan toimittaa testaukseen valituille henkilöille. Tämän jälkeen palveluun syötetään testaajien sähköpostiosoitteet, jotka on liitetty testaajien Windows Live ID -tiliin. Tämä vaihe vaatii, että kaikilla testaajilla on oma Windows Live ID. (Microsoft 2012, Beta Testing Your Application.)



B2B-sovelluksia ei yleensä haluta yleiseen jakeluun. Sovellukset on mahdollista piilottaa yleisestä jakelusta käyttäen App Hubin piilotustoimintoa. Tällöin sovellus ei hyväksynnän jälkeen siirry Marketplaceen, vaan sovellusentekijälle lähetetään linkki, josta sovellus on mahdollista ladata. Tämän linkin sovelluksen tekijä voi puolestaan antaa asiakkaalle. Sovelluksen lataajia ei Marketplacen avulla ole mahdollista autentikoida, vaan kaikki linkin omaavat tahot voivat ladata sovelluksen. Sovellukseen on kuitenkin mahdollista luoda autentikointiominaisuus, jolla laitton käyttö on mahdollista estää. (Microsoft 2012, Targeted Application Distribution.)

#### 4.4 Google

Koska Googlen Android on avoin alusta, sovellusten jakelu sitä käyttäviin laitteisiin on vapaampaa kuin Applella tai Microsoftilla. Google tarjoaa Play-nimisen sovelluskaupan, jonka kautta on mahdollista myydä ja markkinoida sovelluksia keskitetysti. Sovelluskaupan lisäksi kehittäjät voivat jakaa sovelluksiaan esimerkiksi omilla internet-sivuillaan tai sähköpostin välityksellä. Oletuksena Android-laitteet voivat vastaanottaa sovelluksia ainoastaan Play-kaupasta. Jos sovelluksia halutaan asentaa muista lähteistä, täytyy käyttäjän sallia se laitteen asetuksista. (Google 2012, Developers.)

Jos sovelluskehittäjä haluaa myydä maksullisia sovelluksia Play-kaupan kautta, vaatii se rekisteröitymisen ja 25 dollarin rekisteröitymismaksun. Kuten Apple, myös Google tarjoaa mahdollisuuden ansaita sovelluksen sisäisillä ostoilla tai mainoksien avulla. Googlen tarjoama mainospalvelu on nimeltään AdMob. Play-kaupan kautta saaduista tuloista Google tilittää sovelluksen kehittäjälle 70 %. (Google 2012, Developers; Google 2012, Google Play Support.)

## 5 KAJAPRO OY:N INNOVAATIOPROSESSI

Tätä työtä ryhdyttiin tekemään, koska haluttiin selvittää, miten yrityksen innovaatioprosessia olisi mahdollista kehittää eteenpäin. Työn alkaessa yrityksen innovaatioprosessi oli käytännössä asiakaslähtöinen. Tämä tarkoittaa sitä, että ideat ja mahdollinen rahoitus sovellusten kehittämiseen tuli asiakkailta. Sovelluksia myös ideoitiin ja kehitettiin eteenpäin yhdessä asiakkaiden kanssa ja joissain tapauksessa asiakas osallistui myös toteutusvaiheeseen.

Ideoita omiksi sovelluksiksi on yrityksen toiminta-aikana syntynyt jonkin verran. Suurinta osaa omista ideoista ei kuitenkaan ole aktiivisesti ryhdytty viemään eteenpäin. Yksi syy tähän oli, että omien sovellusten kehittämistä varten ei ollut määritelty selkeää innovaatioprosessia. Myös rahoituksen hankkiminen esimerkiksi riskirahoituksen muodossa voi uusille sovelluksille olla hankalaa. Jos rahoittajille voidaan esittää, että sovellusta ryhdytään kehittämään määriteltyjen prosessien mukaisesti, voi tämä helpottaa rahoituksen saamista.

Koska KajaPro on pienehkö ohjelmistoyritys, täytyy innovaatioprosessin kehittämisessä ottaa huomioon sekä yrityksen koko että ohjelmistokehityksen asettamat erikoisvaatimukset. Seuraavissa luvuissa on käyty läpi innovaatioprosessia pienen ohjelmistoyrityksen näkökulmasta sekä esitelty ehdotus KajaPro:n innovaatioprosessiksi.

### 5.1 Innovaatioprosessi pienessä yrityksessä

Innovaatioprosessin käyttämisestä yrityksessä on kiistattomia etuja. Taso, jolla prosessi kannattaa ottaa käyttöön, on yrityskohtaista. Yrityksen koko ja toimiala vaikuttavat suuresti prosessin eri vaiheiden hyödynnettävyyteen. Suuressa yrityksessä on esimerkiksi paremmat mahdollisuudet hyödyntää poikkifunktionaalisuutta innovaatioprosessin eri vaiheissa. Pienessä yhden alan yrityksessä innovaatioprosessin tuominen väkisin osaksi työskentelyä voi turhaan hidastaa projektien etenemistä esimerkiksi lisääntyneen byrokratian kautta. Tässä luvussa eräitä innovaatioprosessin osia on käsitelty tarkemmin erityisesti pienen yrityksen kannalta.

Mahdollisuuksien tunnistaminen on eräs innovaatioprosessin alkupäähän kuuluvista tehtävistä (katso 3.3.1.1). Pienissä yrityksissä samat henkilöt voivat joutua tunnistamaan mahdollisuuksia eri näkökulmista. Tällaisissa yrityksissä esimerkiksi myynti ja markkinointi voivat olla samojen henkilöiden vastuulla ja näiden lisäksi yrityksessä on vain tuotekehitysosasto. Eri asemassa toimivien henkilöiden vähäinen määrä voi vaikeuttaa parhaiden ideoiden löytymistä, koska vuoropuhelua toisten mahdollisuuksia tunnistavien henkilöiden kanssa syntyy vähemmän.

Eräs luvussa 3.3.1.1 esitetty mahdollisuuksien tunnistamiseen käytettävä keino, johon liittyy teknologian kehittymisen seuranta, on erilaiset teknologiafoorumit. Esimerkiksi pienelle ohjelmistoyritykselle teknologiafoorumien järjestäminen tai niihin osallistuminen voi olla haastavaa. Tällaisissa yrityksissä teknologian kehittymisen seurantaan on mahdollista käyttää verkostoitumista yhteisrahoitteisten tutkimushankkeiden kautta. Yliopistoilla ja kehitettävää teknologia hyödyntävillä tahoilla, kuten esimerkiksi sairaaloilla, on yleensä mielenkiintoa olla yhteistyöhankkeissa mukana. Planmedin, joka on suomalainen terveysteknologia-alan yritys, toimitusjohtaja Vesa Mattila kuvaa suomalaista osaamista ja kokemuksiaan yhteistyöhankkeista Ojanperän (2012, 5) lehtiartikkelissa seuraavasti:

*Meillä on pitkät perinteet röntgenlaittevalmistuksesta ja digitaalisesta kuvantamisesta. Yliopistoilla ja sairaaloilla on lisäksi halu olla mukana tutkimus- ja kehitysyhteistyöhankkeissa.*

Henkilöstön oma-aloitteinen seuranta tiettyjen teknologioiden kehittymisestä on myös käyttökelpoinen keino pienelle yritykselle uusien mahdollisuuksien tunnistamiseen. Jos oma-aloitteisuus ei riitä saamaan aikaan tarpeeksi suurta teknologista peittoa, voidaan henkilöille määritellä seurattava teknologia ja seurannalle järjestetään tarpeeksi aikaa muiden töiden rinnalle.

Ideoiden arviointi (katso 3.3.1.4) voi olla haasteellista riippumatta yrityksen koosta. Idean tyyppi ja uutuustaso vaikuttavat kriteerien asettamiseen. Jos idea ja siitä syntyvä tuote ovat niin innovatiivisia, ettei niille ole pystytty määrittelemään selvää asiakastarvetta, täytyy arvioimiseen käytettävät kriteerit määritellä tuotekohtaisesti. Esimerkkejä ohjelmistokehitykseen liittyvistä arviointikriteereistä on esitetty luvussa 5.2.

Poikkifunktionaalisuuden hyödyntämistä painotetaan innovaatioprosessin useassa vaiheessa kuten konseptoinnissa ja tuoteprosessin aikana. Esimerkiksi ohjelmistoalan pienissä yrityksissä poikkifunktionaalisuus jakautuu pääasiassa myyntiin, markkinointiin ja tuotekehitykseen. Tuotekehitysosaston sisällä voi olla erilaista ohjelmistokehitysoasaamista, mutta täysin erisuuntaisten näkemysten saaminen esille voi olla haasteellista. Pienen yrityksen eduksi voidaan ajatella, että tuotekehitysosasto on yleensä suoraan tekemisissä myös asiakkaiden ja muiden sidosryhmien kanssa. Tällöin poikkifunktionaalisia näkemyksiä voidaan saada luontaisesti yrityksen ulkopuolelta. Tuotekehitysosaston kanssakäyminen asiakkaiden kanssa myös tehostaa asiakastarpeiden määrittelyä.

## 5.2 Ohjelmistokehityksen innovaatioprosessi

Ohjelmistokehityksen innovaatioprosessi eroaa jonkin verran yleisestä innovaatioprosessista. Seuraavissa luvuissa käsitellään ohjelmistokehityksen erikoispiirteitä.

### 5.2.1 Prosessin alku

Perinteisen innovaation toteuttaminen, kaupallistaminen ja markkinointi vaativat yleensä suuria taloudellisia panostuksia. Ennen toteutuksen aloittamista täytyy investoida suhteellisen suuria summia laitteistoihin ja toimintaympäristön kehittämiseen sekä mahdolliseen työvoiman palkkaamiseen. Riippuen ohjelmistokehityksen tyypistä suuria panostuksia ei välttämättä prosessin alkuvaiheessa tarvita. Jos kyse on esimerkiksi mobiilisovelluksen kehittämisestä, voi ohjelmistoa päästä toteuttamaan pienellä investoinnilla tai jopa ilmaiseksi, jos työkuluja ei oteta huomioon. Suuren ohjelmistoprosessin aloittaminen voi kuitenkin vaatia laitteisto- ja kehitysympäristöhankintoja sekä yrityksen resurssien ja henkilöstön laajentamista. Tällaisessa tilanteessa rahoituksen hankkiminen projektin toteutusta varten nousee merkittäväksi samoin kuin perinteisissä innovaatioissakin.

Niin kuin aiemmin on mainittu, täytyy uudet ideat arvioida kriittisesti, ennen kuin niitä ryhdytään kehittämään tuotteeksi. Ohjelmistokehitys ei eroa tässä mielessä mekaanisten tuotteiden tai palveluiden kehittämisestä. Ideoiden arviointikriteerit voivat olla samanlaisia hyvin erityyppisissä ideoissa. Kuvasta 4 voidaan nähdä esimerkkejä arviointiin käytettävistä kriteereistä. Kuvan kriteereistä kehityskustannukset, toteutukseen tarvittava aika sekä erityisesti uutuusarvo ovat myös mobiilisovelluksen kannalta oleellisia. Muita kriteerejä erityisesti ohjelmistokehityksessä voivat olla esimerkiksi saatavilla oleva osaaminen toteutusta varten, sovelluksen toteuttamismahdollisuus usealle alustalle sekä kilpailutilanne.

Ohjelmistokehitysprojektin lopputulokseen vaikuttavia määrittelyjä voidaan yleensä muuttaa vielä toteutusvaiheen aikana. Tämä mahdollistaa toteutusvaiheessa keksittyjen ideoiden lisäämisen lopulliseen tuotteeseen. Jossain vaiheessa projektia uusien ideoiden lisääminen kuitenkin täytyy lopettaa. Uudet toiminnot ja ominaisuudet aiheuttavat aina lisää töitä ja hidastavat projektin valmistumista. Kun alkuperäisiin suunnitelmiin pohjautuva ohjelmisto on saatu valmiiksi, voidaan asiakkaan kanssa neuvotella uusien, toteutuksen aikana tulleiden ideoiden lisäämisestä valmiiseen ohjelmistoon. Näin asiakas saa ohjelmiston ensimmäisen version käyttöönsä aikataulussa ja neuvotteluissa voidaan sopia jatkokehityksen vaatimista korvauksista niin, että ne sopivat molemmille osapuolille.

### 5.2.2 Vesiputousmalli

Perinteisesti ohjelmistoprojektien toteutuksessa on käytetty ns. vesiputousmallia. Vesiputousmallissa projekti jaetaan toisiaan lineaarisesti seuraaviin vaiheisiin. Vaiheet ovat yleensä määrittely, suunnittelu, toteutus, testaus ja tuotantoon siirto. (Nykkö & Koskela 2006, 6.)

Vesiputousmallissa on useita ongelmia, jotka vaikuttavat toteutettavan ohjelmiston laatuun, projektin aikataulussa pysymiseen ja projektin lopulliseen hintaan. Koska mallin mukainen eteneminen on lineaarista, ei prosessissa takaisin edelliseen vaiheeseen siirtyminen ole teoriassa mahdollista. Tiukka lineaarisuus voi lisäksi aiheuttaa viiveitä projektiin siten, että seuraavaan vaiheeseen siirtymiseen tarvitaan edellisen vaiheen verifioitu lopputulos. Jokaisen vaiheen lopputuloksena syntyy suuria kokonaisuuksia, usein dokumentteja. Jos projektin sisäinen kommunikointi tapahtuu ainoastaan näiden dokumenttien avulla, altistaa se projektin väärinkäsityksille. Tällainen tehoton kommunikointi huonontaa projektin laatua ja projektin tuloksena voi syntyä ohjelmisto, joka ei enää vastaa liiketoiminnan nykyisiä tarpeita. Tehoton kommunikointi voi myös aiheuttaa sen, että suuri osa aikaisemmissa vaiheissa tapahtuneista väärinkäsityksistä ja virheistä huomataan vasta testausvaiheessa. (Nykopp & Koskela 2006, 6; Jyväskylän yliopisto 2003, 11.)

Kun projektia toteutetaan vesiputousmallin mukaisesti, projektin alussa tehdään yleensä erittäin kattava määrittely. Tämä voi johtaa järjestelmään, johon on varmuuden vuoksi määritetty ominaisuuksia, joita käyttäjät eivät tarvitse. Myös uusien ominaisuuksien lisääminen toteutusvaiheessa voi tiukasti määritellyssä projektissa olla vaikeaa. Muutosten tekeminen vasta projektin loppuvaiheessa vie usein paljon aikaa ja tulee kalliiksi. (Nykopp & Koskela 2006, 6; Jyväskylän yliopisto 2003, 11.)

### 5.2.3 Ketterät menetelmät

Ohjelmistokehitysprojekteille on kehitetty erityisiä ns. ketteriä menetelmiä perinteisen vesiputousmallin korvaajiksi. Tuoteprosessia käsittelevässä luvussa (sivu 34) tärkeimmäksi prosessin ominaisuudeksi mainittiin nopeus. Ketterien menetelmien avulla ohjelmistokehitystä on mahdollista nopeuttaa. Lisäksi kehitettävien ohjelmistojen laatu ja projektin kannattavuus voivat parantua ketterien menetelmien myötä. (Nykopp & Koskela 2006, 7.)

Ketteriin menetelmiin lasketaan sellaiset, jotka suosivat

- ihmisiä ja heidän välistä vuorovaikutusta työkalujen ja prosessien sijasta
- järjestelmän toimivuutta ennen yksityiskohtaisia suunnitelmia
- avointa yhteistyötä tiukkojen sopimusneuvottelujen sijasta
- muutokseen reagoimista aikaisempien suunnitelmien orjallisen seuraamisen sijasta. (Nykopp & Koskela 2006, 7.)

Toisen määritelmän mukaan menetelmä voidaan määritellä ketteräksi, kun se on

- inkrementaalista (julkaisu tehdään pienissä erissä tiheään tahtiin)
- yhteistoiminnallista (kehitysosasto ja asiakas työskentelevät ja kommunikoivat keskenään jatkuvasti)
- suoraviivaista (menetelmä on hyvin dokumentoitu, helppo oppia ja sitä voidaan muokata vaivattomasti)
- mukautuvaa (ohjelmistoon voidaan ottaa mukaan uusia ominaisuuksia toteutuksen aikana). (Abrahamsson, Salo, Ronkainen & Warsta 2002, 17, 98.)

Ketteriä menetelmiä käytettäessä ohjelmisto ja sen toteutus pyritään suunnittelemaan niin, että uusien ominaisuuksien lisääminen on helppoa. Lisäksi ketterät menetelmät lisäävät kanssakäymistä tuotekehityksen ja asiakkaiden välillä. Näin molemmat osapuolet saavat paremman käsityksen siitä, missä mennään ja mitkä ovat vaatimukset. Friman ja Nieminuukko (2010, 24 - 28) mainitsevat nimeltä seuraavat ketterät ohjelmistokehitysmenetelmät:

- Scrum
- Extreme Programming
- Mobile-D
- LEAN.

Ketterien menetelmien tarkempi käsittely on rajattu tämän työn ulkopuolelle.

### 5.3 Esitys innovaatioprosessiksi

KajaPro:lle esitetyn innovaatioprosessin pohjana käytettiin luvussa 3.3 esiteltyä yleistä innovaatioprosessia. Yleistä innovaatioprosessia mukautettiin ohjelmistokehityksen sekä yrityksen koon asettamien erikoisvaatimusten mukaisesti. Esitetyn innovaatioprosessin vaiheet on eritelty kuviossa 2. Seuraavissa luvuissa esitetty innovaatioprosessi on käsitelty vaihe vaiheelta.



KUVIO 2. Esitetyn innovaatioprosessin vaiheet. Innovaatioprosessin alkupään vaiheet on otettu kokonaisuudessaan yleisestä innovaatioprosessista. Konseptoinnin tasoja on yhdistetty.

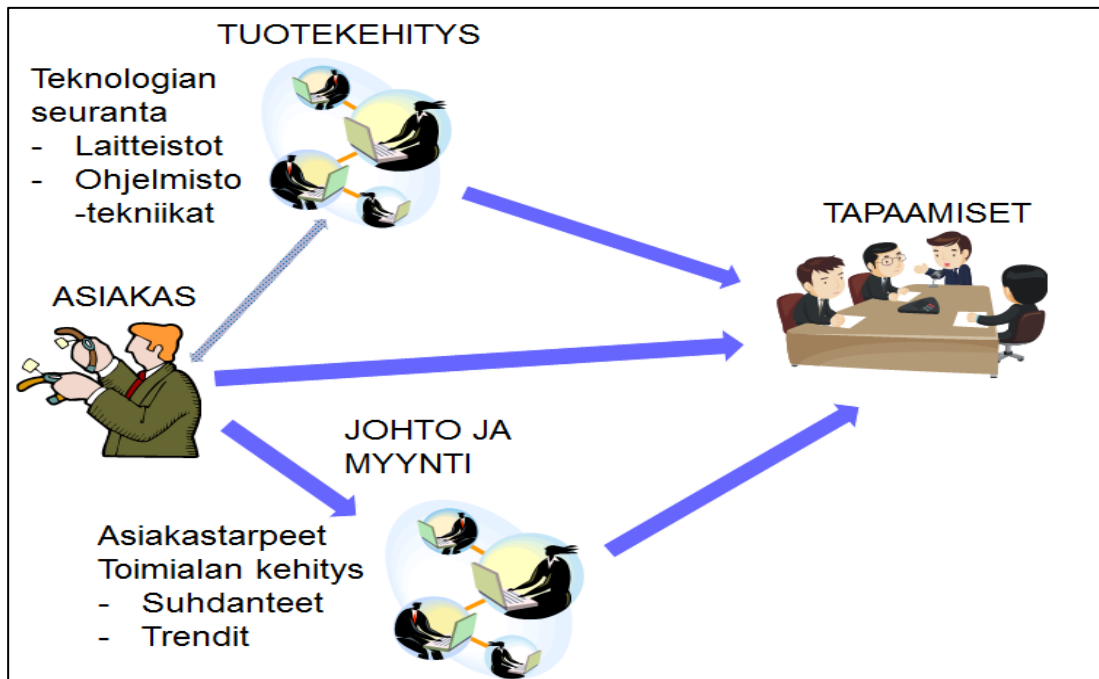
### 5.3.1 Innovaatioprosessin alkupää

Luvussa 3.3.1 esitettyyn innovaatioprosessin alkupäähän kuuluu neljä osa-alueetta: mahdollisuuksien tunnistaminen, ideointi, ideoiden jalostaminen ja ideoiden arviointi. Esitykset näiden osa-alueiden toteuttamiseksi on kuvattu seuraavissa luvuissa.

#### 5.3.1.1 Mahdollisuuksien tunnistaminen

Luvussa 3.3.1.1 kerrottiin, että mahdollisuuksia innovaation tekemiseen voidaan tunnistaa kolmesta eri näkökulmasta: asiakastarve, toimialan mahdollisuudet ja teknologian tarjoamat mahdollisuudet. Mahdollisuuksien tunnistaminen tulisi jakaa yrityksen kaikkien henkilöiden tehtäväksi osaamisalueiden mukaisesti. Esitys mahdollisuuksien tunnistamisen osa-alueiden jakamiseksi on esitetty kuvassa 7.





KUVA 7. Mahdollisuuksien tunnistamisen eri osa-alueiden jakaminen KajaPro:n henkilöstöryhmille. Asiakkaan osallistuminen joka vaiheeseen on avainasemassa.

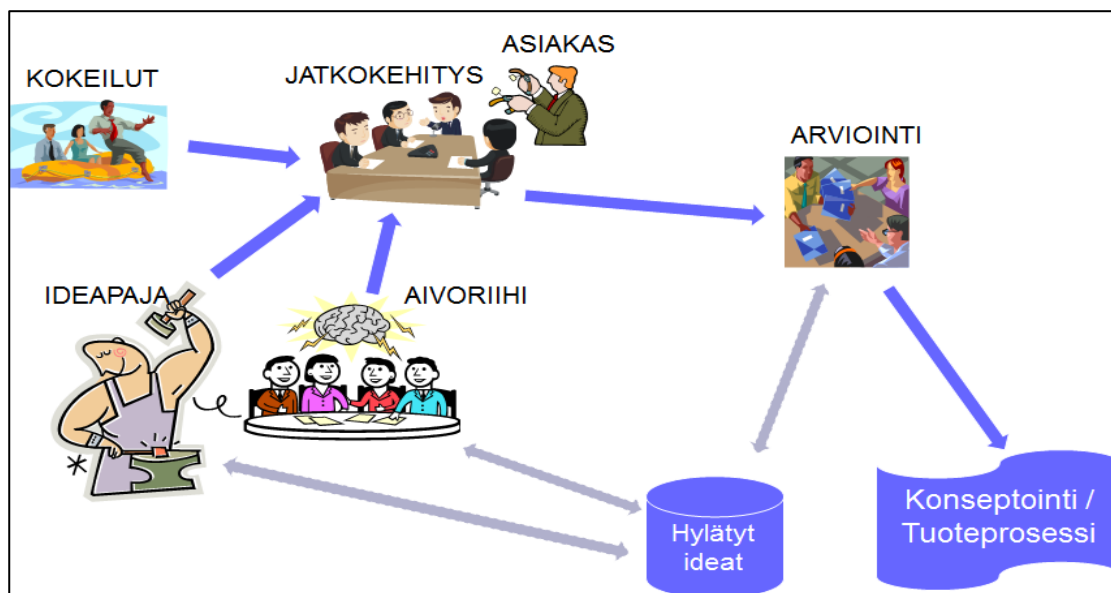
KajaPro:n henkilöstöresurssit huomioon ottaen eri näkökulmia kannattaa yhdistää. Koska johto- ja myyntihenkilöstöllä on työnkuvansa puolesta paremmat näkymät seurata asiakastarpeiden muutoksia ja toimialan kehittymistä, heidän tulisi ottaa päävastuu näiden seurannasta. Tuotekehitysosastolle voidaan antaa päävastuu teknologian mahdollisuuksien seurannasta. Osa tuotekehitysosaston henkilöistä voidaan asettaa seuraamaan pääasiallisesti laitteistopuolella tapahtuvaa kehitystä ja osa ohjelmistotekniikan kehitystä. Vaikka henkilöille on asetettu tietyt seurattavat alueet, tulee heidän peilata oman alueensa mahdollisuuksia myös muiden alueiden avulla. Pienen yrityksen peruspiirteinä myös KajaPro:n tuotekehitysosasto pääsee keskustelemaan suoraan asiakkaiden kanssa. Tämä mahdollisuus kannattaa hyödyntää sekä teknologian seurannassa että asiakastarpeiden tunnistamisessa.

Koska suurin osa yrityksen tuotekehitysosastosta on töissä asiakkaiden projekteissa alihankkijan roolissa, voi työajan käyttäminen mahdollisuuksien tunnistamiseen olla hankalaa. Osa tuotekehityksestä tehdään kuitenkin yrityksen omissa projekteissa. Näille henkilöille mahdollisuuksien tunnistamiseen on helpompi varata aikaa. Mahdollisuuksien tunnistamisen täytyy kuitenkin olla tavoitteellista ja jatkuvaa, joten henkilöiden käyttämää aikaa ja seuranta-aluetta ei tulisi jatkuvasti muuttella.

Mahdollisuuksia tunnistavien henkilöiden yhteiselle tapaamiselle tulisi varata aikaa, jolloin he voivat yhdessä keskustella havainnoistaan. Myös asiakkaiden edustajia tulee ottaa näihin tapaamisiin mukaan tilanteen niin salliessa. Suuremmalla joukolla keskusteltaessa ja asiakkaan mielipide huomioon otettaessa on mahdollista saada uusia näkökulmia omiin havaintoihin. Uudet näkökulmat auttavat saamaan aikaiseksi parempia tuloksia. Aluksi tapaamisia voi olla kerran kuukaudessa noin tunnin verran. Kun tapaamisia on järjestetty joitain kertoja, voidaan tapaamisten pituutta ja tiheyttä muuttaa tarpeiden mukaisesti. Paras toimintatapa tällaisissa tapaamisissa on vapaa keskustelu. Tapaamisissa täytyy kuitenkin varmistua siitä, että kaikki pääsevät ääneen ja, että ainakin osa ajasta käytetään pelkästään uusista havainnoista keskusteluun.

### 5.3.1.2 Ideointi

Ideointi on tärkein yksittäinen osa innovaatioprosessin alkupäästä, koska ilman hyviä ideoita ei synny kaupallisesti kannattavia ohjelmistoja. Kokonaiskuva KajaPro:n innovaatioprosessin ideointiin liittyvistä osa-alueista on esitetty kuvassa 8.



KUVA 8. Ideoinnin eri vaiheet KajaPro:n innovaatioprosessissa. Ideoinnin tuloksena syntyvät ideat etenevät jatkokehityksen kautta arviointiin. Arvioinnissa ideat joko hylätään tai siirretään prosessin seuraaviin vaiheisiin. Asiakkaalla on suuri rooli ideoinnin alkuvaiheissa.

Tasaisin aikaväleihin järjestettävä aivoriihi on helpoin tapa aloittaa yrityksen sisäinen ideointi. Aivoriihiä tulee järjestää useita vuoden aikana. Aluksi järjestysväli voi olla kaksi kuukautta ja sitä voidaan muuttaa tarpeen mukaan. Mahdollisuuksien tunnistamiseen osallistuvien henkilöiden tulee kuulua aivoriihien vakio-osallistujiin. Lisäksi mukaan täytyy ottaa kaikki yrityksen työntekijät, joilla on mahdollisuus sillä hetkellä osallistua. Asiakkaan edustajien osallistuminen aivoriiheeseen on tärkeää. Asiakkaan osallistuminen ideointiin mahdollistaa toimialan ulkopuolisten ideoiden esille tulemisen ja näin parantaa ideoiden yleistä laatua sekä laajentaa ideoiden aihepiiriä.

Kuten luvussa 3.3.1.2 painotettiin, ideoinnin tulee olla kohdennettua. Kohdentaminen ei kuitenkaan saa tukahduttaa ideointia, vaan kaikki esiin tulevat ideat täytyy käsitellä. Esitettävät ideat voivat olla pieniä parannuksia olemassa oleviin tuotteisiin tai esimerkiksi toimintatapoihin. Aivoriihessä on kuitenkin mahdollista esittää täysin uusia ideoita, jotka on havaittu esimerkiksi asiakastarpeiden tunnistuksessa. On myös mahdollista esittää sellaisia ideoita, joille asiakastarvetta ei vielä ole pystytty määrittelemään.

Osallistujilla luultavasti on uusia ideoita esitettäväksi jokaisessa aivoriihessä. Kuten mahdollisuuksien tunnistamiseen liittyvissä palavereissa myös aivoriihessä tulee varmistaa, että kaikki osallistujat pääsevät ääneen. Lisäksi erityisesti ensimmäisissä tapaamisissa tulee vaatia, että jokainen osallistuja esittää ainakin yhden idean.

Aivoriiehtä tukemaan on mahdollista käyttää kuvassa 3 esitettyjä ideoimistapoja. Käytökelpoisia vaihtoehtoja KajaPro:lle ovat kilpailevien tuotteiden analysointi, yrityksen sisäiset ideapajat sekä kokeilut. Vaikka mobiiliympäristössä muutokset ovat nopeita sekä teknologian että ohjelmistojen tasolla, on olemassa olevien ohjelmistojen ominaisuuksien vertailu helppoa. Apuna vertailussa voidaan käyttää luvussa 4 esitettyjä sovelluskauppoja. Kilpailevien sovellusten vertailuun ei kuitenkaan kannata käyttää suuria määriä aikaa.

Yrityksen sisäisiä ideapajoja voidaan järjestää muutaman kerran vuodessa. Näissä ideapajoissa kehitetään uusia ideoita tiettyä kohderyhmää varten. Ideapajoja varten kannattaa varata aikaa enemmän kuin yksittäiseen aivoriiheeseen. Ideapajan sopiva kesto voi olla esimerkiksi yksi tai kaksi työpäivää.

Kokeilujen avulla on mahdollista vastata parhaiten mobiiliympäristön nopeasti muutuviin vaatimuksiin. Kokeiluissa kehitettäviä ideoita on kuitenkin riskialteinta ryhtyä kehittämään eteenpäin. Kokeilujen kautta syntyviä ideoita kannattaa peilata markkinoiden kautta eikä jättää idean menestymistä sattuman varaan. Asiakkaat ovat hyvä tiedonlähde kokeilujen pohjalta kehitettävien ideoiden kohdalla.

Hylätyt ideat kerätään yrityksen sisäiseen tietokantaan myöhempää tarkastelua varten. Ideasta kirjataan ylös keskeisimmät ajatukset sekä hylkäämisen perustelut. Näin kaikki ideat jäävät muistiin ja hylättyjä ideoita ei tarjota uudestaan heti seuraavassa aivoriivessä. Hylättyjä ideoita kannattaa kuitenkin käydä kootusti läpi tietyin väliajoin esimerkiksi aivoriivien ja ideapajojen aikana. Jos jokin idea on hylätty esimerkiksi siksi, että tekniikka ei ole ollut tarpeeksi kehittynyttä idean toteuttamiseksi, on tilanne voinut puolessa vuodessa tai vuodessa muuttua. Muuttuneessa tilanteessa ideaa voidaan ryhtyä kehittämään eteenpäin, jos idea vielä muutoin on kehittämisen arvoisen.

### 5.3.1.3 Ideoiden jatkokehittäminen

Ideoinnin jälkeen hyväksytyt ideat siirretään jatkokehitettäväksi. Kuten yleisen innovaatioprosessin esittelyssä mainittiin, jatkokehitys tulisi ensimmäisessä vaiheessa hoitaa sellaisella ryhmällä, joka ei ole alkuperäisen idean takana. Näin ideaan on mahdollista saada uusia näkökulmia. KajaPro:n ollessa kyseessä voi idean alkuperäisten kehittäjien poissulkeminen jatkokehityksen ensimmäisestä vaiheesta olla hankalaa. Tämä johtuu siitä, että henkilöstöä on rajallinen määrä ja kaikkia ei välttämättä ole mahdollista saada innovaatioprosessin alkupään vaiheisiin mukaan. Toisaalta juuri henkilöitä, jotka eivät aktiivisesti osallistu alkupään toimintoihin, voidaan käyttää jatkokehityksen ensimmäisessä vaiheessa hyödyksi. Jos idean kehittämiseen on mahdollista saada apua joltain ideaan liittyvältä sidosryhmältä, kuten esimerkiksi asiakkailta, kannattaa tämä ulkopuolinen apu käyttää hyödyksi.

Jatkokehityksen aikana ei ole tarpeen saada aikaiseksi täysin valmista ideaa. Idean kehittäminen eteenpäin jatkuu konseptointivaiheessa, jossa asiakastarpeita ja idean toteuttamiseen liittyviä asioita ratkotaan tarkemmin. Jatkokehityksen tarkoitus on jalostaa idea niin hyväksi, että sillä on mahdollisuudet päästä ideoiden arvioinnista konseptointivaiheeseen. Tämän vuoksi ideoiden jatkokehittämiseen käytettävät resurssit ja aika tulee mitoittaa järkevästi.

Ehdotus on, että hyväksytyt ideat jaetaan kahden hengen ryhmille alustavaa jatkokehitystä varten. Nämä ryhmät käyttävät jatkokehitykseen aikaa maksimissaan kaksi tuntia ideaa kohden. Kun kaikki ideat on käyty läpi, kokoonnutaan suuremmalla joukolla viimeistelemään jatkokehitys. Viimeistelyyn käytetään muutama tunti riippuen ideoiden määrästä. Lopuksi yrityksen johtoryhmän täytyy arvioida ideat luvussa 3.3.1.3 esitettyjen kriteerien pohjalta. Päätösten avuksi voidaan toteuttaa yksinkertainen prototyypisovellus, jonka avulla mahdollisia toiminnallisuuksia voidaan esitellä. Prototyyppiä voidaan käyttää apuna ja kehittää eteenpäin prosessin seuraavissa vaiheissa.

#### 5.3.1.4 Ideoiden arviointi

Jatkokehityksen jälkeen ideat täytyy arvioida kriittisesti ja päättää, ryhdytäänkö ideaa toteuttamaan. Idean toteutukseen on mahdollista hankkia rahoitusta ulkopuolisilta tahoilta. Viimeistään tässä vaiheessa täytyy päättää, aletaanko idean toteutusta varten hakea uutta rahoitusta vai pyritäänkö idea toteuttamaan olemassa olevalla rahoituksella.

Tunnistettu asiakastarve voi auttaa riskirahoituksen hankkimisessa. Ideaa ei kuitenkaan kannata hylätä sillä perusteella, että varmoja markkinoita valmiille sovellukselle ei vielä ole. Mobiilisovellusten markkinoilla hyvillä uusilla ideoilla on mahdollista saada merkittävä tuotto. Jos sovellusta ryhdytään kehittämään eteenpäin, siirretään se konseptointivaiheeseen. Olemassa olevaan projektiin liittyvästä innovaatiosta täytyy päättää, liitetäänkö idea osaksi projektia. Resurssien ja rahoituksen riittäminen toteutusta varten täytyy arvioida tarkasti. Jos idea liitetään osaksi olemassa olevaa projektia, ei uutta konseptointivaihetta tarvita.

#### 5.3.2 Konseptointi

Kaikki luvussa 3.3.3 esitetyt konseptointitasot ovat hyödyllisiä KajaPro:n innovaatioprosessin kannalta. Tällä hetkellä rajalliset resurssit kuitenkin pakottavat joko valitsemaan yhden tai kaksi tärkeintä tasoa, joita tullaan käyttämään jatkuvasti. Toinen mahdollisuus on yhdistellä eri konseptointitasoja. Tässä luvussa konseptointitasojen keskeiset ominaisuudet on kerrattu lyhyesti ja valitaan KajaPro:n innovaatioprosessiin ehdotettavat tasot.

Visioiva konseptointi on mobiilisovelluksia kehittävälle yritykselle tärkeää, koska siinä pyritään näkemään asiakkaiden tulevaisuuden tarpeita sekä lyhyellä että pitkällä aikavälillä. Koska visioiva konseptointi on jatkuva prosessi, vaatii se ainakin yhden henkilön jatkuvan työpanoksen. Konseptointia ei kuitenkaan tarvitse toteuttaa kokopäiväisesti. Visioiva konseptointi toimii yhteistyössä ideoiden jatkokehittämisen kanssa. Ideoiden jatkokehityksessä mahdollisesti luotava esimerkkisovellus auttaa tekemään konseptointiin liittyviä päätöksiä ja konseptoinnin tulokset tukevat jatkokehitystä.

Seuraavaa konseptoinnin tasoa, kehittävä konseptointia, on yrityksessä toteutettu jo aiemmin. Erityisesti kehittävä konseptointiin liittyvät asiakastarpeiden kerääminen valituilta asiakkailta sekä erilaisten teknologisten ratkaisujen kehittäminen ovat olleet käytössä. Kehittävä konseptointi on jatkuva prosessi, joten sekin tarvitsee jatkuvan työpanoksen. Jos kehittävä konseptointi halutaan ottaa mukaan innovaatioprosessiin, kannattaa miettiä, voidaanko se yhdistää visioivan konseptoinnin kanssa. Jos kehittävä konseptointi halutaan pitää omana prosessinaan, voidaan se antaa samojen henkilöiden hoidettavaksi kuin visioiva konseptointikin.

Määrittelevä konseptointi on tärkeää, koska se vaikuttaa suoraan valmistuvaan sovellukseen sekä sen toteutustapaan. Määrittelevän konseptoinnin aikana asiakastarpeet pyritään muuttamaan tuoteominaisuuksiksi ja mietitään, miten sovelluksesta on mahdollista saada paras mahdollinen tuotto. Sovellukseen liittyvät palvelut voivat esimerkiksi olla parempi tulojen lähde, kuin pelkällä sovelluksen myynnillä on saatavissa. Palveluja voivat olla esimerkiksi sovelluksen avulla kerätyn tai tuotetun datan käsittely ja säilytys. Ideoiden karsimista määrittelevän konseptoinnin aikana tulee välttää, jos resurssien puute ei siihen pakota. Jos näyttää siltä, että idean toteuttamiseen ei ole tällä hetkellä resursseja, voidaan se jättää odottamaan resurssien vapautumista ja toteuttaa myöhemmin.

Ohjelmistoprojektissa ratkaiseva konseptointi kannattaa toteuttaa pääosin tuoteprosessin kanssa rinnakkain. Ennen toteutuksen aloittamista asiakastarpeet, ratkaisut ja käytettävät tekniikat kannattaa yhdistää selkeäksi kokonaisuudeksi. Tämän jälkeen aloitetaan tuoteprosessi määrittelyjen mukaisesti. Ratkaisevan konseptoinnin avulla voidaan sovellukseen tuoda uusia, toteutuksen aikana havaittuja tarpeita mukaan toteutukseen. Samalla voidaan tarkkailla, täyttävätkö toteutetut ominaisuudet tarkoituksensa.

Tärkeimmät esitetyistä konseptointitasoista KajaPro:n kannalta ovat visioiva ja ratkaiseva konseptointi. Nämä konseptointitasot tulee ottaa kokonaisuudessaan mukaan yrityksen innovaatioprosessiin. Kehittävä konseptointi kannattaa liittää osaksi visioivaa konseptointia. Yhdistäminen kannattaa tehdä resurssisyistä ja siksi, että kehittävään konseptointiin liittyvät vaiheet ovat jo yritykselle tuttuja. Näihin vaiheisiin on valikoitunut sellaisia henkilöitä, jotka pystyvät hoitamaan myös visioivan konseptoinnin. Jos kehittävä konseptointi halutaan myöhemmässä vaiheessa muuttaa kokonaan itsenäiseksi, voidaan se antaa samojen henkilöiden hoidettavaksi kuin visioiva konseptointikin. Jos määrittelevän konseptoinnin jättää prosessista pois, voi hyppäys visioivasta konseptoinnista ratkaisevaan konseptointiin olla liian suuri. Ehdotus onkin, että määrittelevä konseptointi otetaan alkuvaiheessa mukaan ja myöhemmin tarkastellaan, onko siitä mahdollista luopua.

### 5.3.3 Tuoteprosessi

Pienessä ohjelmistokehitykseen keskittyneessä yrityksessä tuoteprosessin etukäteiseen suunnitteluun ei kannata käyttää paljon resursseja. Jos projekti on sellainen, että sen suorittaa muutama henkilö, ei tuoteprosessia kannata hidastaa lisäämällä siihen ylimääräistä byrokratiaa. Suuremmissa usean hengen projekteissa, jotka kestävät useita kuukausia, jokin ketteristä menetelmistä kannattaa ottaa toteutustavaksi, koska ne parantavat ohjelmiston laatua ja mahdollistavat tuoteprosessin aikaisen rahallisen hyödyn. Ketterän menetelmän käyttöönotto täytyy päättää ennen projektin aloittamista, koska se vaikuttaa huomattavasti projektin suunnitteluun.

### 5.3.4 Palkitsemiskäytännöt

Palkitsemiseen kannattaa pyrkiä käyttämään kertakorvausta. Kuten luvussa 3.3.2 mainittiin, tämä auttaa tuomaan esille parhaat ideat. Rahat korvausta varten voidaan ottaa esimerkiksi yrityksen innovaatorahastosta tai projektin budjettiin lasketaan tämä kertakorvaus mukaan ja se tilitetään idean keksijöille esimerkiksi palkan mukana. Pelkistä hyvistä ideoista ei kuitenkaan kannata palkita suoraan vaan vasta sitten, kun ideasta aletaan kehittää sovellusta.

## 6 INNOVAATIOPROSESSIN TESTAAMINEN KÄYTÄNNÖSSÄ

Työn aikana oli tarkoitus tutkia Windows Phonen tarjoamia mahdollisuuksia esitetyn innovaatioprosessin avulla ja kehittää sovellus tälle alustalle. Samalla arvioitiin sovelluksen tarjoamia mahdollisuuksia ja tutkittiin ansaintamalleja sovelluksen kaupallistamista varten. Tässä luvussa on käsitelty innovaatioprosessin testaamista sekä sovelluksen toteuttamista.

### 6.1 Mahdollisuuksien tunnistaminen ja ideointi

Innovaatioprosessi aloitettiin mahdollisuuksien tunnistamisella. Mahdollisuuksien tunnistamiseen kuului Windows Phonen käyttöliittymän tarjoamien ominaisuuksien tutkiminen. Lisäksi tuli tutkia HTML5:n ja JavaScriptin tarjoamia mahdollisuuksia. HTML5 otettiin tutkittavaksi, koska se tarjoaa mahdollisuuden luoda alustariippumattomia sovelluksia. Alustariippumattomuus on eräs tärkeimmistä KajaPro:ssa käytettävien tekniikoiden ominaisuuksista.

Samaan aikaan mahdollisuuksien tunnistamisen aikana aloitettiin toteutettavan sovelluksen ideointi. Ideointiin käytettiin muutaman hengen aivoriihiä. Aivoriihissä syntyi ideoita esimerkiksi erilaisista liikuntaharrastuksen seurantaan tarkoitetuista sovelluksista. Näistä ideoista mitään ei kuitenkaan tässä vaiheessa ryhdytty kehittämään. Yksi syy on jo aiemmin mainittu Bluetooth-yhteyden rajoitettu käytettävyyys (katso 2.6.3). Bluetooth-yhteyden rajoitukset vaikeuttavat esimerkiksi sykemittareiden liittämistä sovellukseen.

Lopulta päätettiin, että Windows Phonen käyttöliittymän testaamiseen toteutettaisiin sovellus KajaPro:ssa aiemmin ideoidun sovelluksen pohjalta. Sovellus on nimeltään Probill. Se on laskupohjan luomiseen ja eteenpäin lähettämiseen tarkoitettu sovellus, joka oli aiemmin toteutettu sekä Windows-ympäristöön että Androidille. Windows Phone -sovelluksen toiminnallisuutta ja käyttöliittymää oli mahdollista vertailla aiemmin toteutettujen versioiden kanssa. Vertailu helpottaa eri alustojen ominaisuuksien ja rajoitusten huomioonottamista suunniteltaessa sovelluksia tulevaisuudessa.



## 6.2 HTML5:n tarjoamat mahdollisuudet

C- ja C++-koodien yhteensopivuus käytettävän alustan kanssa on tärkeää KajaPro:lle, koska suurin osa yrityksen sovelluksista on toteutettu näillä ohjelmointikielillä. Yhteensopivuus helpottaa olemassa olevien sovellusten siirtämistä uudelle alustalle. Koska mahdollisuuksien tunnistamisen aikana ei ollut varmaa tietoa, milloin Windows Phone tulee tarjoamaan yhteensopivuuden em. ohjelmointikielien kanssa, ryhdyttiin asiaa tutkimaan HTML5:n ja JavaScriptin kautta.

Yrityksessä oli evaluoitu työkalua, jolla C/C++-koodi pystyttiin muuttamaan HTML5 / JavaScript-koodiksi. Työkalu on nimeltään Emscripten (Kripken 2012). Tämän vuoksi keskityttiin selvittämään, millaisia työkaluja tai ohjelmistokehyksiä HTML5-ohjelmien kirjoittamiseksi Windows Phonelle on tarjolla. Toimivien työkalujen avulla C/C++-koodit olisi mahdollista siirtää Windows Phonelle ilman suurta panostusta uudelleen ohjelmointiin.

Tutkimusten avulla löydettiin Adoben ja Apachen yhteishallinnassa oleva PhoneGap/Cordova. Se on ohjelmistokehys, jonka avulla on mahdollista päästä käsiksi useiden eri mobiililaitteiden sisäisiin ohjelmointirajapintoihin käyttäen HTML- ja JavaScript-koodeja. Tämä mahdollistaa esimerkiksi laitteiden eri antureiden käytön. PhoneGap/Cordova on yhteensopiva Windows Phonen kanssa, ja se integroituu osaksi Windows Phone SDK:ta. (Adobe 2012; Apache 2012.)

## 6.3 Sovellus

Probill-sovelluksen toteuttamiseen ei tarvinnut määritellä erillistä tuoteprosessia, vaan sovellus toteutettiin pääasiassa innovaatioprosessin alkupään aikana. Toteutuksessa käytettiin Microsoftin tarjoamia työkaluja ja sovelluksen testaamiseen ja virheiden etsintään käytettiin KajaPro:n hankkimaa Samsung Omnia 7 -älypuhelinta, joka oli rekisteröity sovellusten kehittämistä varten. Liitteessä 2 on esitelty toteutuneen sovelluksen käyttöliittymää.

## 6.4 Innovaatioprosessin tulokset

Toteutetun sovelluksen avulla saatiin selvitettyä Windows Phonen käyttöliittymän mahdollisuuksia. Käyttöliittymän ulkonäkö ja toiminnallisuus vakuuttivat KajaPro:n henkilökunnan siitä, että myös Windows Phonelle on mahdollista tehdä samantyyppisiä sovelluksia kuin yrityksessä on aiemmin tehty esimerkiksi Androidille.

Mahdollisuuksien tunnistamisen aikana selvisi, että Windows Phone 8 tulee Microsoftin mukaan sisältämään tuen C/C++-kielille (Belfiore, 2012). Tämän vuoksi KajaPro:ssa jäädään odottamaan Windows Phone 8:n virallista julkistusta eikä sovelluksia ryhdytä aktiivisesti kehittämään Windows Phone 7:lle. Windows Phone 8:n tarjoama C/C++-tuki riittää KajaPro:n tarpeisiin tällä hetkellä, joten nykyisiä sovelluksia ei muuteta HTML5-muotoon vain sen takia, että ne saataisiin toimimaan Windows Phone 7 -alustassa.

Vaikka kaikkia työn aikana kehitettyjä ideoita ja löydettyjä tekniikoita ei ryhdytä aktiivisesti kehittämään tai oteta välittömästi käyttöön, on ne prosessin mukaisesti laitettu muistiin. Näin niitä voidaan tarpeen mukaan käyttää omassa tuotekehityksessä tai asiakkaiden projekteissa.

## 7 YHTEENVETO JA POHDINTA

Työn tavoitteena oli luoda esitys yrityksen uudeksi innovaatioprosessiksi sekä tutkia Windows Phone -alustan tarjoamia mahdollisuuksia yrityksen ohjelmistokehitykselle. Molemmat tavoitteet saavutettiin työn aikana.

Työn tuloksena yritykselle syntyi innovaatioprosessi, jota ryhdytään mahdollisuuksien mukaan ottamaan käyttöön. Koska asiakaslähtöisyys oli vahvasti osana yrityksen innovoimista jo ennen työn aloittamista, tulee se säilyttää tärkeänä osana myös uudessa innovaatioprosessissa. KajaPro:ssa on myös huomattu, että erilaisten teknisten ratkaisujen jatkuva innovointi on tärkeää asiakasneuvotteluja ajatellen. Asiakasneuvotteluissa on helpompi esittää valmiita ratkaisumalleja asiakkaan ongelmiin, kun käytettävissä on useita eri tekniikkaa käyttäviä vaihtoehtoja. Kehitetyn innovaatioprosessin tarkoitus on tukea myös tällaista innovoimista.

Tarkasti määritelty innovaatioprosessi hyödyttää eniten suuria yrityksiä, joilla on mahdollisuuksia hyödyntää yrityksen sisäistä multifunktionaalisuutta prosessin eri vaiheissa. Pienelle yritykselle innovaatioprosessi mahdollistaa omien ideoiden kehittämisen määritetyn prosessin avulla. Tämä voi auttaa ideoiden saamisessa tuotteeksi ja markkinoille. Innovaatioprosessi vaatii kuitenkin resursseja ja johdon sitoutumista prosessiin. Pienessä yrityksessä täytyy miettiä tarkasti, milloin ja missä suuruudessa innovaatioprosessi kannattaa ottaa käyttöön.

Innovaatioprosessista on saatavilla paljon tietoa sekä kirjallisuudesta että internetistä. Ongelmaksi muodostui oikeiden, helposti ymmärrettävien tietolähteiden valinta. Kun oikeat lähteet saatiin valittua, oli innovaatioprosessin teoriaan perehtyminen melko nopeaa. Haasteellisinta oli oikeasti höydyllisen tiedon erottelu kaikesta tarjolla olevasta ja tämän erotellun tiedon tiivistäminen luettavaan muotoon. Haastetta lisäsi se, ettei innovaatioprosessin teoria työn vaatimassa laajuudessa ollut tuttua aiemmista opinnoista tai työelämästä.

Innovaatioprosessin teoriaan perehtymisen jälkeen siirryttiin tutkimaan eri mobiilialustoihin sekä mobiilisovellusten jakeluun ja markkinointiin liittyviä asioita. Tämä hidasti päivitetyn innovaatioprosessin kehittämistä, koska osa innovaatioprosessiin liittyvistä asioista täytyi kerrata innovaatioprosessiin palattaessa. Parempi vaihtoehto olisi ollut päivitetyn innovaatioprosessin kehittäminen heti innovaatioprosessin teoriaan perehtymisen jälkeen. Esitys yrityksen innovaatioprosessiksi kuitenkin valmistui suunnittelussa ajassa, joten suurta haittaa asioiden kertaamisesta ei lopputuloksen kannalta ollut.

Jos innovaatioprosessia käyttöönotettaessa havaitaan, että jotkin osat esitetystä prosessista eivät käy yritykselle tai esitetystä prosessista puuttuu joitain oleellisia osioita, kannattaa prosessi päivittää tarvittavilta osilta. Päivityksen jälkeen innovaatioprosessia kannattaa testata yrityksen oikeassa toimintaympäristössä. Päivitys ja testaus voidaan toteuttaa esimerkiksi osana toista opinnäytetyötä.

Työn toinen konkreettinen tulos oli innovaatioprosessin alkupäätä tukemaan luotu sovellus. Sovellus saatiin pääosin valmiiksi innovaatioprosessin testaamisen aikana. Sovelluksen viimeistely ja myyntiin laittaminen jäivät tehtäväksi työn valmistumisen jälkeen.

Vaikka Windows Phone -sovellusten toteuttamisesta oli hyvin vähän aikaisempaa kokemusta, onnistui se kuitenkin melko vaivattomasti. Sovelluksen toteuttamisessa auttoi, että internetistä löytyi vastaus jokaiseen ohjelmoinnissa kohdattuun ongelmaan. Sovelluskehitykseen käytetyt työkalut ja niiden toiminnot olivat pääosin tuttuja, koska Windows-sovelluskehityksessä käytetään samoja työkaluja. Tutut työkalut auttoivat pitämään sovelluskehityksen aikataulussa. Ainoastaan Expression Blend 4 oli täysin uusi työkalu. Sen käyttö oli helppo opetella internetistä löytyvien resurssien avulla.

Suurin osa sovellukseen määritellyistä ominaisuuksista saatiin mukaan jo tässä vaiheessa. Muutama sovelluksen asetuksien muuttamiseen liittyvä ominaisuus jäi sovelluksesta puuttumaan. Puuttuvien ominaisuuksien lisäksi sovelluksen jatkokehitykseen jäi lähinnä käytettävyyden ja käyttöliittymän hiominen kuntoon. Nämä muutokset täytyy tehdä, jos sovellus aiotaan laittaa myyntiin. Päätöstä myyntiin laittamisesta ei vielä tämän työn päättymiseen mennessä ollut tehty.

## LÄHTEET

Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. 2002. *Agile software development methods: Review and analysis* [verkkoasiakirja]. VTT publications 478 [viitattu 26.5.2012]. Saatavissa: <http://www.vtt.fi/inf/pdf/publications/2002/P478.pdf>.

Adams, S. 2011. Dilbert. 37. 2012 nro 7. 19.

Adobe. *PhoneGap* [verkkosivu]. [viitattu 25.9.2012]. Saatavissa: <http://phonegap.com/>.

Amy. *A Brief History of Windows Mobile* [verkkoasiakirja]. [viitattu 17.9.2012]. Saatavissa: <http://notebooks.com/2010/04/12/a-brief-history-of-windows-mobile>.

Apache. *Cordova* [verkkosivu]. [viitattu 25.9.2012]. Saatavissa: <http://incubator.apache.org/cordova/>.

Apilo, T. & Taskinen T. 2006. *Innovaatioiden johtaminen* [verkkoasiakirja]. VTT tiedotteita 2330 [viitattu 2.5.2012]. Saatavissa: <http://www.vtt.fi/inf/pdf/tiedotteet/2006/T2330.pdf>.

Apilo, T., Taskinen T. & Salkari, I. 2007. *Johda innovaatioita*. Helsinki: Talentum.

Appcelerator / IDC. 2012. *Q1 2012 Mobile Developer Report* [verkkoasiakirja]. Saatavissa: <http://www.appcelerator.com/company/survey-results/mobile-developer-report-q-2012/>.

Apple. *Developer* [verkkosivu]. [viitattu 6.2.2012]. Saatavissa: <https://developer.apple.com/>.

BBC. 2012. *Blackberry-maker RIM plans new focus amid \$125m loss* [verkkosivu]. [viitattu 12.4.2012]. Saatavissa: <http://www.bbc.co.uk/news/business-17557177>.

Belfiore, J. *Announcing Windows Phone 8* [verkkosivu]. [viitattu 25.9.2012]. Saatavissa: [http://windowsteamblog.com/windows\\_phone/b/windowsphone/archive/2012/06/20/announcing-windows-phone-8.aspx](http://windowsteamblog.com/windows_phone/b/windowsphone/archive/2012/06/20/announcing-windows-phone-8.aspx).

Blueberry Consultants. *Software Development For The Mobile Market* [verkkosivu]. [viitattu 17.11.2012]. Saatavissa: <http://www.bbconsult.co.uk/Mobile-Web-Software-Development.aspx>.

Bohn, D. *iOS: A visual history* [verkkosivu]. [viitattu 17.11.2012]. Saatavissa: <http://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>.

Cooper, R. C. 1993. *Winning at new products. Accelerating the process from idea to launch*. 2nd ed. Cambridge: Perseus Books.

Cooper, R. C. 1999. *Product leadership. Creating and launching superior new products*. Cambridge: Perseus Books.

Dundon, E. 2002. *The seeds of innovation. Cultivating the synergy that fosters new ideas*. New York: American Management Association.

eNorssi-yhteistyöverkoston www-sivu [verkkosivu]. [viitattu 2.5.2012]. Saatavissa: <http://www.enorssi.fi/>.

Fogelholm, C-M. 2009. *Tuoteideasta innovaatioksi: Tuoteideoiden ja keksintöjen kaupallistaminen suomalaisessa innovaatiojärjestelmässä*. Tampere: Mediapinta.

Friman, J. & Nieminuukko, J. 2010. *Ketterien ohjelmistokehitysmenetelmien soveltaminen innovaatioprosessiin* [verkoasiakirja]. Lappeenrannan teknillinen yliopisto. Teknistaloudellinen tiedekunta. Tuotantotalouden osasto. Kandidaatintyö. Saatavissa: <http://www.doria.fi/bitstream/handle/10024/69556/nbnfi-fe201104291491.pdf?sequence=3>.

Gartner. *Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent* [verkkosivu]. [viitattu 6.2.2012]. Saatavissa: <http://www.gartner.com/it/page.jsp?id=1848514>.

Google. *Developers* [verkkosivu]. [viitattu 6.2.2012]. Saatavissa: <http://developer.android.com/index.html>.

Google. *Google Play Support* [verkkosivu]. [viitattu 27.9.2012]. Saatavissa: <http://support.google.com/googleplay>.

Heikkilä, J. 2010. *Luovasta ideasta innovaatioon: Luovuus ja innovatiivisuus selvitysmiskeinona*. Turku: Enostone.

Jaakola, J. 2011. *Nokia valitsi Microsoftin - riittääkö se* [verkkosivu]? Suomenkuva-lehti. [viitattu 28.3.2012]. Saatavissa: <http://suomenkuvalehti.fi/jutut/talous/nokia-valitsi-microsoftin-riittaako-se>.

*Kansallinen innovaatiostrategia*. 2008 [verkkoasiakirja]. Saatavissa: [https://www.tem.fi/files/19704/Kansallinen\\_innovaatiostrategia\\_12062008.pdf](https://www.tem.fi/files/19704/Kansallinen_innovaatiostrategia_12062008.pdf).

Kissinger, A. *How to Make an HTML5 iPhone App* [verkkosivu]. 26.3.2010 [viitattu 6.2.2012]. Saatavissa: <http://sixrevisions.com/web-development/html5-iphone-app/>.

Kokkonen, V., Kuuva, M., Leppimäki, S., Lähteinen, V., Meristö, T., Piira, S. & Sääskilähti, M. 2005. *Visioiva tuotekonseptointi. Työkalu tutkimus- ja kehitystoiminnan ohjaamiseen* [verkkoasiakirja]. Teknoliigiateollisuuden julkaisuja 4/2005. Helsinki: Teknoliigainfo Teknova. Saatavissa: [http://www.teknoliigainfo.net/content/kirjat/pdf-tiedostot/Alkusivut\\_Visioiva\\_tuotekonseptointi.pdf](http://www.teknoliigainfo.net/content/kirjat/pdf-tiedostot/Alkusivut_Visioiva_tuotekonseptointi.pdf).

Kollanus, S. *Ohjelmistotekniikka, prosessimallit* [verkkoasiakirja]. Jyväskylän yliopisto. [viitattu 26.5.2012]. Saatavissa: <http://users.jyu.fi/~kolli/ohjelmistotekniikka/files/prosessimallit.pdf>.

Korkiakangas, J-P. *Lisenssin osto ja hankintapäätökseen vaikuttavat tekijät* [verkkoasiakirja]. Jyväskylän yliopisto. Taloustieteellinen osasto. Taloustieteen pro gradu - tutkielma. Saatavissa: <https://jyx.jyu.fi/dspace/bitstream/handle/123456789/8481/1005.pdf?sequence=1>.

Korpela, J. 2008. *Pienehkö sivistyssanakirja* [verkkosivu]. [viitattu 8.5.2012]. Saatavissa: <http://www.cs.tut.fi/~jkorpela/siv/index.html>.

Kotilainen, S. 2012. *Tässä on Nokia Lumia 900 – suuri, ohut lte-puhelin* [verkkosivu]. Tietokone. [viitattu 22.4.2012]. Saatavissa: [http://www.tietokone.fi/uutiset/tassa\\_on\\_nokia\\_lumia\\_900\\_suuri\\_ohut\\_lte\\_puhelin](http://www.tietokone.fi/uutiset/tassa_on_nokia_lumia_900_suuri_ohut_lte_puhelin).

Kripken. *Emscripten* [verkkosivu]. [viitattu 1.10.2012]. Saatavissa: <https://github.com/kripken/emscripten>.

Kuittinen, T. *How to Create a \$1M App with a \$10K Marketing Budget* [verkkosivu]. Forbes. [viitattu 3.6.2012]. Saatavissa: <http://www.forbes.com/sites/terokuittinen/2012/02/01/how-to-create-a-1m-app-with-a-10k-marketing-budget/>.

Lappalainen, E. 2011. *Nyt tulee Lumia tupaan! Nokia julkisti ensimmäiset Windows-puhelimet* [verkkosivu]. Talouselämä. [viitattu 22.4.2012]. Saatavissa: <http://www.talouselama.fi/nokialandia/nyt+tulee+lumia+tupaan+nokia+julkisti+ensimmaiset+windowspuhelimet/a2083111>.

Lavonen, J. & Meisalo, V. *Luovan ongelmanratkaisun työtavat* [verkkosivu]. Helsingin yliopisto. [viitattu 2.6.2012]. Saatavissa: <http://www.edu.helsinki.fi/malu/kirjasto/lor/index.htm>.

Lehto, T. 2012. *Nokia Lumia 900 kerää myönteisiä arvioita* [verkkosivu]. 3T. [viitattu 22.4.2012]. Saatavissa: [http://www.3t.fi/artikkeli/uutiset/teknologia/nokia\\_lumia\\_900\\_keraa\\_myonteisia\\_arvioita](http://www.3t.fi/artikkeli/uutiset/teknologia/nokia_lumia_900_keraa_myonteisia_arvioita).

Microsoft. *App Hub* [verkkosivu]. [viitattu 25.3.2012]. Saatavissa: <http://create.msdn.com/en-US/>.

Microsoft. *Beta Testing Your Application* [verkkosivu]. [viitattu 25.3.2012]. Saatavissa: <http://msdn.microsoft.com/en-us/library/hh334585%28v=vs.92%29.aspx>.

Microsoft. *Expression* [verkkosivu]. [viitattu 15.4.2012]. Saatavissa: <http://www.microsoft.com/expression/>.

Microsoft. *Marketplace* [verkkosivu]. [viitattu 25.3.2012]. Saatavissa: <http://www.windowsphone.com/fi-fi/store>.

Microsoft. *MSDN Library* [verkkosivu]. [viitattu 15.4.2012]. Saatavissa: <http://msdn.microsoft.com/en-us/library/ms123401.aspx>.

Microsoft. *Targeted Application Distribution* [verkkosivu]. [viitattu 25.3.2012]. Saatavissa: <http://msdn.microsoft.com/en-us/library/hh334584%28v=vs.92%29.aspx>.

Microsoft. *Visual Studio* [verkkosivu]. [viitattu 15.4.2012]. Saatavissa: <http://www.microsoft.com/visualstudio/eng/team-foundation-service>.



Nirvi, N. *Remedyn ralli ohitti 10 miljoonaa* [verkkosivu]. Pelit. [viitattu 4.6.2012]. Saatavissa: [http://www.pelit.fi/uutiset/590/Remedyn\\_ralli\\_ohitti\\_10\\_miljoonaa](http://www.pelit.fi/uutiset/590/Remedyn_ralli_ohitti_10_miljoonaa).

Nokia *Developer Wiki* [verkkosivu]. [viitattu 26.9.2012]. Saatavissa: [http://www.develo-per.nokia.com/Community/Wiki/Wiki\\_Home](http://www.develo-per.nokia.com/Community/Wiki/Wiki_Home).

Nykopp, S. & Koskela, L. 2006. *Projektien laadun parantaminen ketterillä menetelmillä* [verkkoasiakirja]. Systemityö. [viitattu 26.5.2012]. 2006 nro 3. 6 - 8. Saatavissa: <http://www.pcuf.fi/sytyke/lehti/kirj/st20063/ST063-06A.pdf>.

Ojanperä, V. 2012. Terveyttä Suomesta. *3T*. 2012 nro 16. 5.

Perttula, M. & Sääskilahti, M. 2004. *Product Concept Development as a Concious Resource. NordDesign 2004 - Product Design in Changing Environment*. 18.-20.8.2004. Tampere. Jaottelu on esitetty myös kirjassa Keinonen, T., Jääskö, V. (toim.) 2004. *Tuotekonseptointi*. Teknologiaateollisuuden julkaisuja 12/2004. Helsinki: Teknologiainfo Teknova.

Rapeli, K. 2012. *Nokian Elop: Lumian myyntitulokset vaihtelevia* [verkkosivu]. Arvopaperi. [viitattu 22.4.2012]. Saatavissa: <http://www.arvopaperi.fi/uutisarkisto/nokian+elop+lumian+myyntitulokset+vaihtelevia/a801374>.

Research In Motionin www-sivu [verkkosivu]. [viitattu 12.4.2012]. Saatavissa: <http://www.rim.com/>.

Sinerma, O. 2012. Peliala suurten muutosten kourissa. *Pelit*. 2012 nro 3. 62.

Tuominen, T. 2010. *Käytettävyys- ja käyttöliittymäsuunnittelu mobiililaitteelle* [verkkoasiakirja]. Tampereen ammattikorkeakoulu. Tietotekniikan koulutusohjelma. Ohjelmistotekniikka. Opinnäytetyö. Saatavissa: [https://publications.theseus.fi/bitstream/handle/10024/14023/Tuominen\\_Terhi.pdf?sequence=2](https://publications.theseus.fi/bitstream/handle/10024/14023/Tuominen_Terhi.pdf?sequence=2).

Viswanathan, P. *What is a Mobile Device* [verkkosivu]? [viitattu 10.4.2012]. Saatavissa: <http://mobiledevices.about.com/od/glossary/g/What-Is-A-Mobile-Device.htm>.

Vähimaa, A. 2012. Toisinajattelija: Samsung Wave III. *MikroBitti*. 2012 nro 4. 62.



## ESIMERKKEJÄ WINDOWS PHONE -SOVELLUSKEHITYKSEEN KÄYTETTÄVISTÄ TYÖKALUISTA

The screenshot displays the Microsoft Visual Studio 2010 Express for Windows Phone environment. The main window shows the source code for `MainPage.xaml.cs` in the `Probill_wp7` project. The code defines a `PhoneApplicationPage` class with a constructor that initializes `settings` as an instance of `IsolatedStorageSettings.ApplicationSettings`. A comment indicates that the current culture is set to the culture associated with the selected locale.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.Globalization;
using System.Threading;
using System.IO.IsolatedStorage;

namespace Probill_wp7
{
    public partial class MainPage : PhoneApplicationPage
    {
        // Constructor
        public MainPage()
        {
            var settings = IsolatedStorageSettings.ApplicationSettings;
            CultureInfo newCulture;
            // set this thread's current culture to the culture associated with the selected locale
            if (settings.Contains("SelectedCulture"))
            {
            }
        }
    }
}

```

The Error List at the bottom shows one warning:

Errors	Warnings	Messages	Description
0	1	0	The variable 'ex' is declared but never used

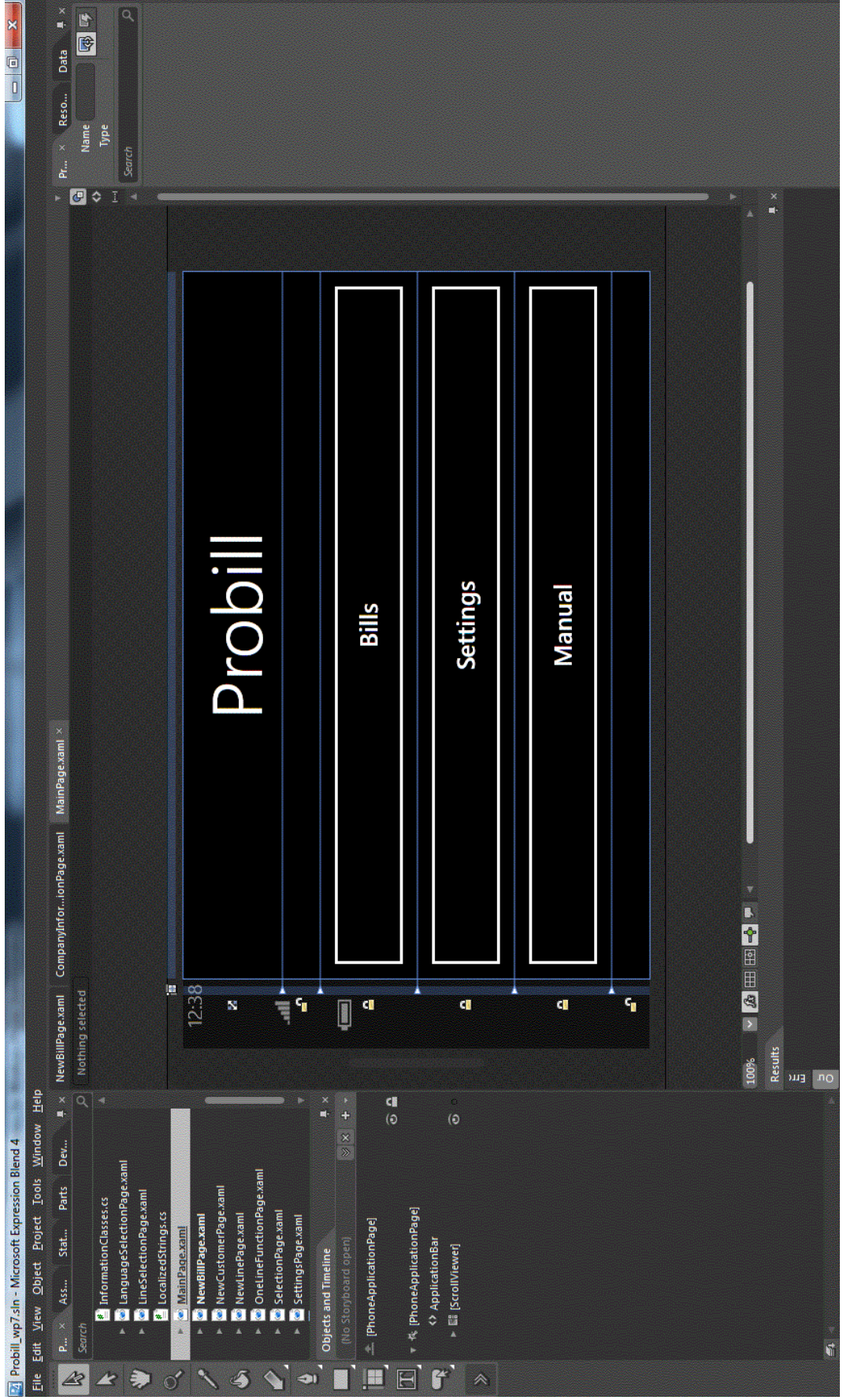
The Solution Explorer on the right shows the project structure, including files like `App.xaml.cs`, `App.xaml`, `ApplicationIcon.png`, `AppResources-en-GB.res`, `AppResources-fi-FI.res`, `AppResources.resx`, `Background.png`, `BillSelectionPage.xaml`, `CompanyInformationPa`, `InformationClasses.cs`, `LanguageSelectionPage`, `LineSelectionPage.xaml`, `LocalizedStrings.cs`, `MainPage.xaml`, and `NewBillPage.xaml`.

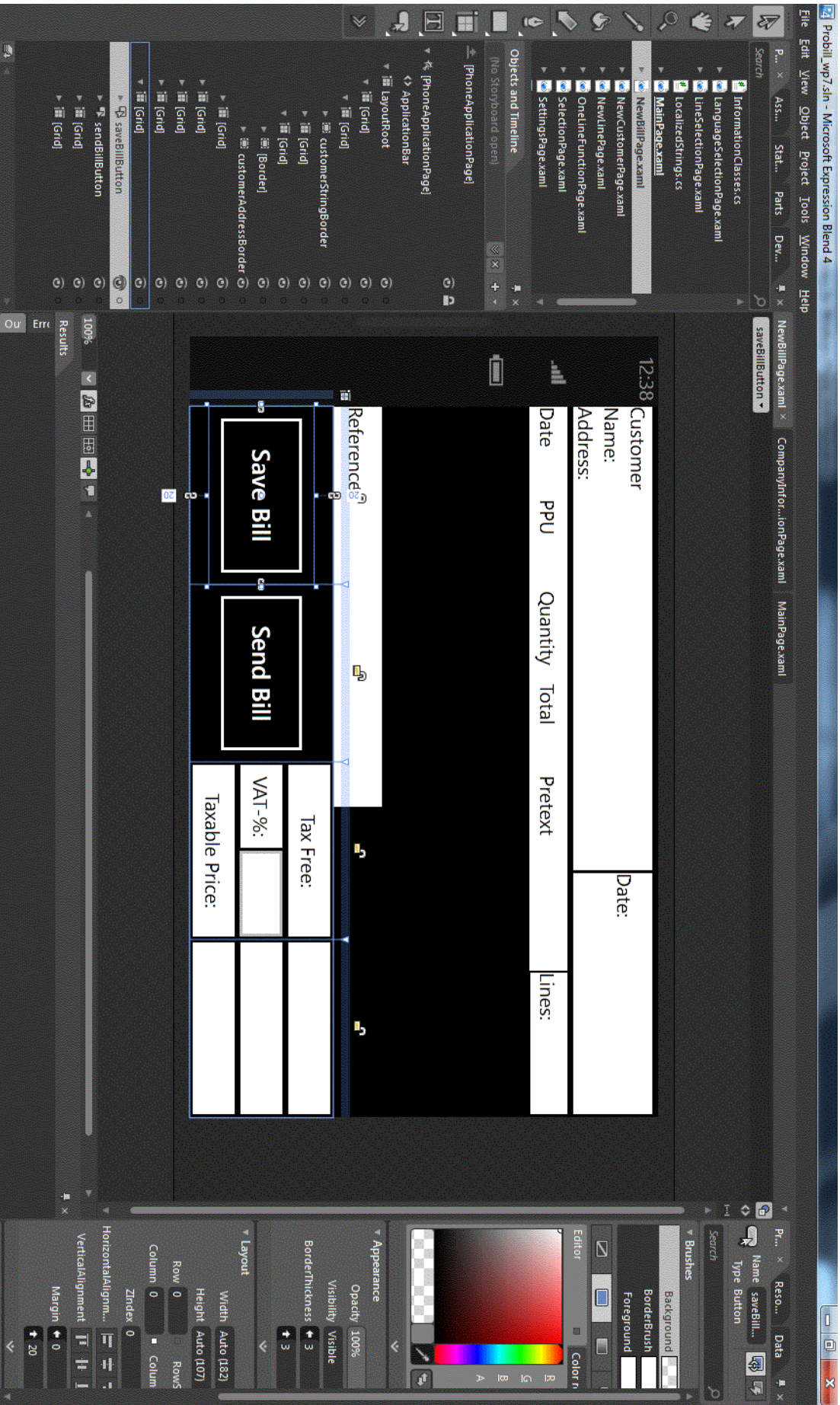
The screenshot displays the Microsoft Visual Studio 2010 Express for Windows Phone environment. The main window is divided into several panes:

- Top Pane:** Shows the mobile application interface. It features a dark background with the text "Probill" at the top. Below it are three menu items: "Bills", "Settings", and "Manual".
- Code Pane:** Displays the XAML code for the application. The code defines a root grid with a title panel and a text block.
 

```

<!--layoutRoot is the root grid where all page content is placed-->
<Grid x:Name="layoutRoot" Background="Transparent">
  <ScrollViewer>
    <Grid x:Name="layoutRoot" Background="Transparent">
      <RowDefinition Height="Auto"></RowDefinition>
      <RowDefinition Height="*"></RowDefinition>
      <RowDefinition Height="100"></RowDefinition>
      <RowDefinition Height="100"></RowDefinition>
      <RowDefinition Height="100"></RowDefinition>
      <RowDefinition Height="*"></RowDefinition>
    </Grid.RowDefinitions>
  </ScrollViewer>
  <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="3">
    <TextBlock x:Name="pageTitle" Text="Probill" VerticalAlignment="C
      
```
- Properties Pane:** Located at the bottom right, it shows the properties for the selected element, "DataContent".
- Solution Explorer:** Located at the bottom left, it shows the project structure, including folders for "App.xaml", "App.xaml.cs", "AppResources", "AppResources-en-GB.res", "AppResources-fi-Fin.res", "AppResources-resx", "Background", "Background.png", "BillsSelectionPage.xaml", "CompanyInformationPage.xaml", "InformationClasses", "LanguageSelectionPage.xaml", "LineSelectionPage.xaml", "LocalizedStrings", "MainPage.xaml", "MainPage.xaml.cs", "MainPage.xaml.cs", and "NewBillPage.xaml".
- Error List:** Located at the top left, it shows a single error: "The variable 'ex' is declared but never used".
- Toolbox:** Located at the top right, it shows various UI controls available for the application.





### TOTEUTETUN WINDOWS PHONE -SOVELLUKSEN NÄYTTÖKUVIA

