



15.4.2010

Open your mind. LUT.
Lappeenranta **University of Technology**

TEKNISTALOUDELLINEN TIEDEKUNTA
TUOTANTOTALOUDEN OSASTO
CS90A0050 Kandidaatintyö ja seminaari

Ketterien ohjelmistokehitysmenetelmien soveltaminen innovaatioprosessiin

Applying agile methods to innovation process

Kandidaatintyö

Joel Friman
Jyri Niemimuukko

TIIVISTELMÄ

Tekijät: Joel Friman, Jyri Niemimuukko	
Työn nimi: Ketterien ohjelmistokehitysmenetelmien soveltaminen innovaatioprosessiin Applying agile methods to innovation process	
Osasto: Tuotantotalous	
Vuosi: 2010	Paikka: Lappeenranta
Kandidaatintyö. Lappeenrannan teknillinen yliopisto. 39 sivua, 2 taulukkoa ja 14 kuvaa Tarkastaja: Tutkijatohtori Lea Hannola	
Hakusanat: Innovaatioprosessi, Ketterät menetelmät, Innovaatioprosessin alkupää, Ohjelmistokehitys	
Keywords: Innovation Process, Agile Methods, Fuzzy Front-End of Innovation, Software Development	
<p>Kandidaatin työmme tarkoitus on tutkia voisiko ketteriä ohjelmistokehitysmenetelmiä soveltaa innovaatioprosessiin. Innovaatioprosessissa on hyvin samankaltaiset vaiheet kuin perinteisessä ohjelmistokehitysprosessissa eli vesiputousmallissa. Vesiputousmallin ongelmia on ratkaistu ketterien menetelmien keinoin, joten päätimme lähteä tutkimaan löytyisikö perinteisen ohjelmistokehitysprosessin ja innovaatioprosessin ongelmista yhteneviä piirteitä. Oletimme, että jos ongelmat yhtenevät tarpeeksi, voisi ketteristä menetelmistä löytyä keinoja innovaatioprosessin ongelmien ratkaisemiseen. Tavoitteena on löytää innovaatioprosessin parannuskohteita, joihin ketteristä menetelmistä löytyisi ratkaisu joko kokonaan tai osittain.</p> <p>Aluksi tarkastellaan innovaatioprosessin historiaa, kehitystä ja ongelmia. Innovaatioprosessin tutkimisen jälkeen käydään läpi järjestelmäkehitysprosessin historiaa, kehitystä ja ongelmia. Järjestelmäkehitysprosessin jälkeen käydään läpi merkittävimmät ketterät menetelmät, jotka ovat ratkaisseet perinteisen ohjelmistokehitysprosessin keskeisiä ongelmia.</p> <p>Työmme lopussa on innovaatioprosessin ja järjestelmäkehitysprosessin synteesi. Synteesissä tutkitaan ongelmien yhtenevyyttä ja ketterien menetelmien ratkaisuja ongelmiin. Synteesissä myös pohdimme, mitä ketterien menetelmien työkaluista voisi soveltaa innovaatioprosessin.</p> <p>Työn johtopäätös on, että ketteristä menetelmistä löytyy sovellettavia työtapoja innovaatioprosessiin. Työn laajuus estää soveltamisen mahdollisuuksien tarkemman tutkimisen ja yksittäisten ketterien työtapojen soveltaminen osaksi innovaatioprosessia olisikin hyvä jatkotutkimuksen aihe.</p>	

SISÄLLYSLUETTELO

1	Johdanto	1
2	Innovaatioprosessi.....	3
2.1	Innovaatio	3
2.2	Yleinen prosessi	5
2.3	Kehitys ja sukupolvet.....	7
2.3.1	1G: Teknologiahyönteinen sukupolvi	10
2.3.2	2G: Markkinavetoinen sukupolvi	11
2.3.3	3G: Interaktiivinen sukupolvi	12
2.3.4	4G: Toiminnot integroiva sukupolvi.....	13
2.3.5	5G: Järjestelmät integroiva verkottunut sukupolvi.....	14
3	Nykyaikaisen innovaatioprosessin ongelmat.....	16
3.1	Alkupää ja konseptointi	17
3.2	Tuoteprosessi	18
4	Järjestelmäkehitysprosessi	20
4.1	Vesiputousmallin historia ja kehitys	20
4.2	Vesiputousmallin ongelmat	22
5	Ketterät menetelmät	23
5.1	Scrum	24
5.2	Extreme programming	25
5.3	Mobile-D.....	26
5.4	LEAN.....	27
6	Innovaatioprosessin ja järjestelmäkehitysprosessin synteesi.....	29
6.1	Alkupää	31
6.2	Epävarmuuden ja muutosten hallinta.....	31
6.3	Raskas dokumentaatio	32
6.4	Tiedon siirtäminen	33
7	Johtopäätökset.....	35
8	Yhteenveto	38
	Lähteet.....	40
	Liitteet	

1 Johdanto

Innovaatioprosessissa on hyvin samankaltaiset vaiheet kuin perinteisessä ohjelmistokehitysprosessissa eli nk. vesiputousmallissa. Vesiputousmallin käytännön soveltaminen on kuitenkin osoittautunut ongelmalliseksi sen kaavamaisuuden ja joustamattomuuden takia. Näiden ongelmien korjaamiseksi mallia on muunneltu ja varioitu lukuisin eri tavoin ja viimeisimpänä kehitysaskelena 2000-luvun alussa on muodostunut joukko menetelmiä, joita yhdistää pyrkimys nopeaan asiakasarvon ja käyttökelpoisen ohjelmistoversion tuottamiseen vaiheittaisen ohjelmistokehitysmallin avulla. Näitä menetelmiä kutsutaan ketteriksi ohjelmistokehitysmenetelmiksi.

Innovaatioprosessi kattaa laajemman kokonaisuuden kuin perinteinen tuotekehitysprosessi. Tuotekehitysprosessin lisäksi innovaatioprosessi sisältää alkupään, konseptointivaiheen sekä mahdollisen tutkimusosuuden. Innovaatioprosessin ongelmat löytyvät alkupäästä, jota pidetään sumeana tai kaaosmaisena, sekä konseptoinnista ja tuotekehityksestä. Alkupäätä kuvataan usein nimellä ”fuzzy front-end of innovation”. Innovaatioprosessi on kehittynyt yksinkertaisista lineaarisista prosesseista monimutkaisiksi interaktiivisiksi verkostomalleiksi, mutta samalla alkupään, konseptointivaiheen ja tuotekehitysprosessin uudet ongelmat ovat muodostuneet merkittäviksi. Monimutkainen prosessirakenne vahvistaa alussa tehtyjä ”virheitä”. Nykyaikaisen innovaatioprosessin ongelmat liittyvät alkupäähän, konseptointivaiheeseen ja tuoteprosessiin.

Työn keskeinen tutkimuskysymys on se, että voidaanko innovaatioprosessissa hyödyntää tietojärjestelmien kehittämisessä hyväksi havaittuja ketteriä menetelmiä joko kokonaan tai osittain. Kysymykseen vastaaminen edellyttää ohjelmistokehitys- ja innovaatioprosessien analysointia niiden keskinäisten erojen ja yhtäläisyyksien osalta sekä niihin kohdistuvan kritiikin vertailua.

Työn tavoitteena on löytää innovaatioprosessin parannuskohteita joihin ketterillä ohjelmistokehitysprosesseilla tai niistä lainatuilla yksittäisillä menetelmillä voitaisiin vaikuttaa prosessin tehokkuutta parantavalla tavalla. Mikäli kritiikki tai prosessien ongelmakohdat yhtenevät ja ketteristä menetelmistä löytyy ongelmiin ratkaisuja, voidaan olettaa ratkaisujen sopivan myös innovaatioprosessiin.

Työssä ei käsitellä tyhjentävästi kaikkia innovaatioprosessien muotoja, vaan keskitytään niiden kehityksen tai kritiikin kannalta oleellisimpiin malleihin. Työ ei myöskään pyri innovaatioprosessin

ja perinteisen ohjelmistokehitysprosessin synteisiin eikä siinä tarkastella vesiputousmallin muita muunnelmia tai ketterien menetelmien ja vesiputousmallin ulkopuolelle jääviä ohjelmistokehitysmenetelmiä. Myös palveluinnovaatioprosessin tutkiminen, sekä innovaatioprosessin lanseeraus- ja markkinointivaiheet on jätetty tarkastelun ulkopuolelle, jotta pysytään annetussa laajuudessa.

Tutkimustuloksiin ja johtopäätöksiin pyritään löytämällä innovaatioprosessin ja ketterien ohjelmistokehitysmenetelmien yhteisiä ominaisuuksia, tavoitteita ja ongelmia. Mikäli yhteisiä tekijöitä ja erityisesti yhteisiä ongelmia löytyy ratkaisuihin, oletamme ratkaisujen olevan sovellettavissa ohjelmistokehityksestä innovaatioprosessiin.

2 Innovaatioprosessi

Määritellään ennen varsinaisen innovaatioprosessin tutkimista innovaation käsite, jotta innovaatioprosessin tutkiminen olisi mielekästä. Innovaation käsitteen määrittelyn jälkeen käydään ensin innovaatioprosessi yleisesti läpi, jonka jälkeen tutustutaan innovaatioprosessin eri sukupolviin ja kehitysvaiheisiin.

2.1 Innovaatio

Terminä innovaatio tulee latinankielisestä sanasta ”Innovare” tarkoittaen ”jonkin uuden luomista”. Innovaatio määritellään prosessiksi, jossa mahdollisuudet muovataan uusiksi ideoiksi ja onnistutaan tuomaan kaupalliseen tai yhteiskunnalliseen käyttöön. (Tidd, Bessant & Pavitt 2005, s. 66)

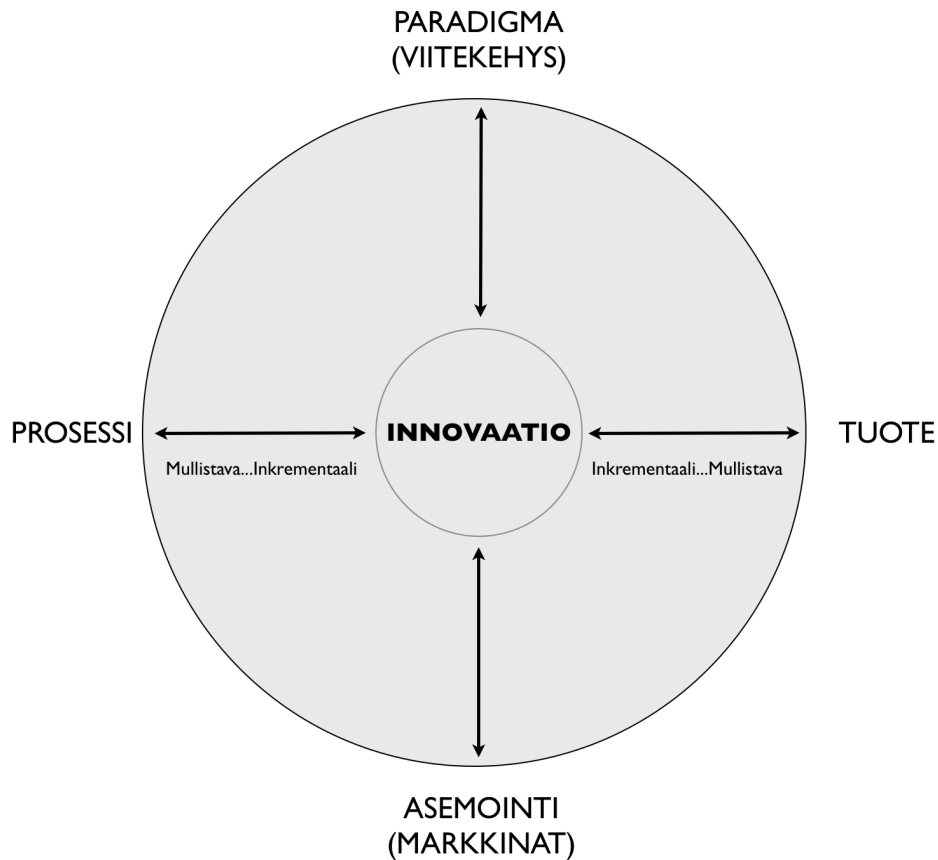
UK:n Innovaatioyksikkö määrittelee innovaation uuden idean menestyksekkäänä hyödyntämisenä. Hyödyntämisellä painotetaan sitä, että pelkästään uuden idean keksiminen ei riitä innovaatioon, vaan idea täytyy myös pystyä tuomaan markkinoille. (Innovation Unit UK, 2004) Rothwellin ja Gardnieren mukaan innovaation ei tarvitse olla suuri teknologinen edistysaskel (radikaali innovaatio), vaan pienetkin edistysaskeleet teknologisessa tietotaidossa voivat olla innovaatiota (inkrementaalinen innovaatio). (Rothwell & Gardnier 1985, s. 168)

Tilastokeskus on määritellyt innovaation markkinoille tuoduksi uudeksi tai oleellisesti parannetuksi tuotteeksi tai uudeksi tai olennaisesti parannetuksi tuotantomenetelmäksi. Innovaatiot perustuvat joko uuteen teknologiaan, entisen teknologian soveltamiseen uudella tavalla tai uuden tiedon hyödyntämiseen. (Tilastokeskus 2005)

Innovaatiot voidaan jakaa neljään erilaiseen ryhmään luonteensa perusteella, jolloin muodostuu Innovaatioavaruus, joka on esitetty kuvassa 1. Innovaatioavaruuden neljä eri ulottuvuutta on esitetty alla;

- **Tuoteinnovaatio** on muutos tuotteissa (materiaalisissa tai immateriaalisissa), joita yritys tarjoaa. Tuoteinnovaatiota on esimerkiksi auton uudenlainen muotoilu.
- **Prosessi-innovaatio** on muutos tuotanto- ja jakelumenetelmissä, joilla tuote tuotetaan ja toimitetaan.

- **Asemointi-innovaatio** (Position innovation) on muutos asiayhteydessä, jossa tuotteet esitetään, eli muutos tuotteen markkinointikontekstissa.
- **Paradigma-innovaatio** on muutos, joka muuttaa kuluttajien ajattelutapaa tai yrityksen toiminnan viitekehystä. (Tidd et al. 2005, s. 10)



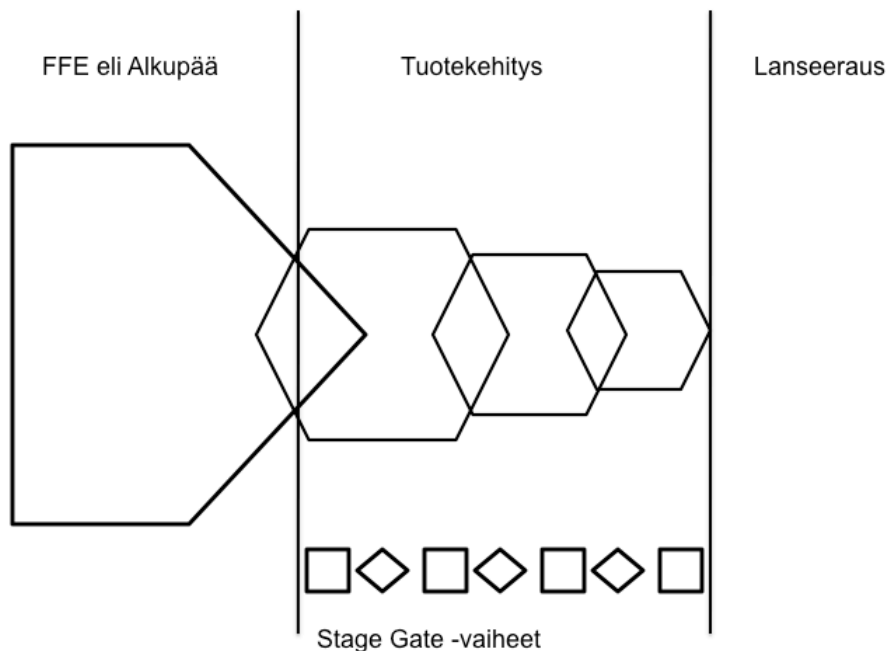
Kuva 1. Innovaatioavaruus. (Tidd et al. 2005, s. 13)

Kuvassa 1 näkyvä kaksisuuntainen nuoli, kuvaa innovaation aikaansaaman muutoksen vaikutusta. Inkrementaalinen muutos on vähäinen lisäys tuotteeseen tai prosessiin, joka parantaa vanhaa tuotetta tai prosessia itsessään uudella tavalla. Radikaali muutos sen sijaan tuottaa tuotteen tai palvelun, joka on kokonaisuudessaan uudenlainen. On myös muistettava, että havaittu uutuusaste on suhteellinen. Isolle ohjelmistoyritykselle jokin arkinen tekninen parannus voikin olla pienelle ohjelmistoyritykselle mullistava. (Tidd et al. 2005, s.11)

2.2 Yleinen prosessi

Karkealla tasolla kaiken tyyppisille innovaatioprosesseille on löydettävissä samat vaiheet eli innovaatioprosessin alkupää ja varsinainen toteutusvaihe. Ennen varsinaista projektointia toteutusvaihe alkaa konseptoinnilla ja esisuunnittelulla tai molempia käyttäen. Varsinainen toteutusvaihe koostuu tuotekehitys- ja testausvaiheesta. Vaiheet ovat joko lineaarisia eli toisiaan seuraavia tai iteratiivisia eli toistuvia, jolloin ne eivät etene suoraviivaisesti. (Apilo, Taskinen & Salkari 2007, s 131)

Kuvassa 2 on esitetty nykyaikaisen innovaatioprosessin jako alkupäähän, tuotekehitykseen ja tuotekehityksen jälkeisiin markkinointitoimenpiteisiin. Tuotekehitysvaiheessa näkyy ”Stage Gate” -portit, joilla tuotekehityksen tulosta varmennetaan koko prosessin ajan.



Kuva 2. Nykyaikaisen innovaatioprosessin jako alkupäähän, tuotekehitykseen ja lanseeraukseen. (Kahn 2005, s. 82)

Verrattuna perinteiseen tuote- tai tuotekehitysprosessiin, innovaatioprosessi sisältää laajemman kokonaisuuden. Tarkkaan määritellyn projektiosuuden lisäksi innovaatioprosessi sisältää innovaatioprosessin alkupään (fuzzy front-end) ja mahdollisen tutkimusosuuden. Nykyaikaista innovaatioprosessia ei voida pitää pelkän tuotekehitysosaston asiana, vaan siihen liittyy useita yrityksen muita toimintoja. (Apilo et al. 2007, s. 132)

Innovaatioprosessin alkupää tarjoaa suurimmat haasteet, mutta myös samalla suurimmat mahdollisuudet onnistumiseen. Alkupään merkitys on suuri, koska siinä muodostetaan käsitys esimerkiksi teknologioiden, markkinoiden ja asiakkaiden tarpeiden kehittymisestä tulevaisuudessa. Alkupäässä yritys valitsee tulevan kilpailukyvyn takaamiseen tarvittavien innovaatioiden kehittämiseen vaadittavat idut. (Apilo & Taskinen. 2006, s. 43)

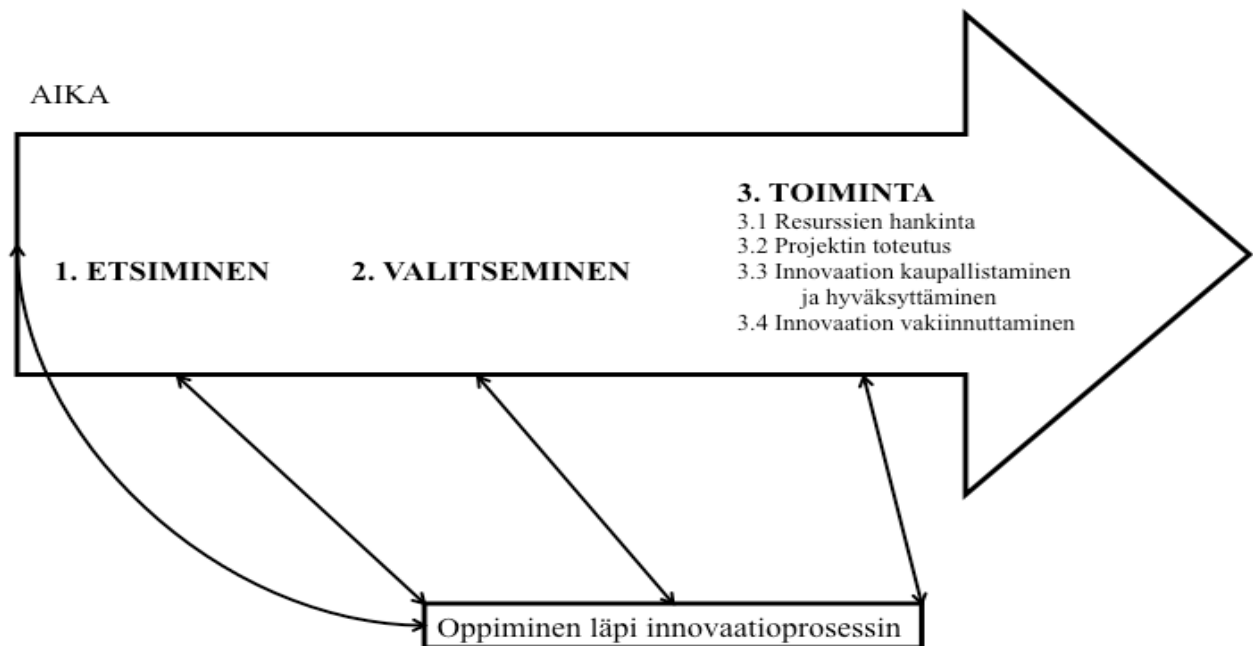
Innovaatioprosessi voidaan nähdä ydinprosessina organisaation sisällä. Innovaatio on yleistä toimintaa, joka liittyy vahvasti yrityksen selviämiseen ja kasvuun. Jokaisen yrityksen on pystyttävä uusiutumaan pystyäkseen kannattavaan liiketoimintaan, joten innovaatioprosessin ydintä voidaan pitää yleisenä kaikille yrityksille. (Tidd et al. 2005, s. 67)

Yksinkertainen innovaatioprosessin perusmalli voidaan esittää kolmen toisiaan seuraavan vaiheen avulla;

1. **Etsiminen** on yrityksen sisäisen ja ulkoisen ympäristön tarkkailua ideoiden (signaalien) varalta, jotta voidaan tunnistaa uhat ja mahdollisuudet, jotka mahdollistavat muutoksen.
2. **Valitseminen** on vaihe, jossa päätetään minkä ideoiden suhteen aiotaan toimia.
3. **Toimintavaiheen** tarkoituksena on valittujen ideoiden potentiaalinen muutos uudeksi tuotteeksi tai palveluksi ja sen lanseeraaminen sisäisillä ja/tai ulkoisilla markkinoilla.

Toimintavaihe jaetaan neljään *alaprosessiin*, jotka seuraavat toisiaan järjestyksessä;

- 3.1 *Tiedollisten resurssien hankinta* innovaation mahdollistamiseksi. Esimerkiksi uuden tuotekehitysratkaisun kehittäminen tai osaavien henkilöiden rekrytointi on tiedollisten resurssien hankintaa.
- 3.2 *Projekti toteutetaan*, usein epävarmoissa olosuhteissa, jonka takia projektiryhmältä vaaditaan laajaa ongelmanratkaisukykyä.
- 3.3 *Innovaatio julkaistaan ja julkaisun jälkeen pyritään hallitsemaan kuluttajien hyväksymisprosessia.*
- 3.4 *Vakiinnutetaan innovaatio ja sen käyttö pitkällä tähtäimellä – tai palataan alkuperäiseen ideaan ja muovataan sitä vastaamaan kysyntään paremmin. Koko innovaatioprosessin ajan tulisi toimintaa arvioida, jotta oppiminen olisi mahdollista. (Tidd et al. 2005, s. 67)*
Innovaatioprosessin yksinkertainen perusmalli on esitetty kuvassa 3.



Kuva 3. Innovaatioprosessin yksinkertainen perusmalli. (Tidd et al. 2005, s. 67)

2.3 Kehitys ja sukupolvet

Innovaation syntymisen ja ilmenemisen ymmärtämiseksi on kehitetty useita teorioita, jotka kaikki tarkastelevat innovaatioprosessia eri näkökulmista teorian kehittämishetken trendien mukaisesti. Rothwell on jakanut nämä teorit viiteen historialliseen sukupolveen. Rothwellin jako sukupolviin on esitetty taulukossa 1. (Rothwell 1994, s.7)

Taulukko 1. Innovaatioprosessien jako sukupolviin Rothwellin mukaan. (Rothwell 1994, s.7)

Sukupolvi	Ominaista
Ensimmäinen ja toinen	Yksinkertaisia lineaarisia malleja: markkinoiden veto ja teknologian työntö.
Kolmas	Kytetty malli: vuorovaikutuksen tunnistaminen ja palautesilmukoiden käyttö prosessissa.
Neljäs	Rinnakkainen malli: toimintojen integrointi, pääpaino yhteistyössä.
Viides	Järjestelmien integrointi ja korkea verkostoitumisen taso läpi prosessin, joustava ja räätälöity vastaus. Jatkuva innovaatio.

Varhaiset mallit (1. ja 2. sukupolvi) tarkastelevat innovaatiota lineaarisena prosessina. Innovaatioprosessi jakautuu toisiaan seuraaviin toiminnollisiin vaiheisiin. Tieteellisen ja teknologisen tutkimuksen tuottamat sovellukset luovat mahdollisuuden innovatiiviselle tuotteelle, joka tuodaan markkinoille - tätä kutsutaan teknologiatyöntöiseksi innovaatioprosessiksi (technology push). Toisaalta markkinoilta lähtöisin oleva tarvesignaali jollekin uudelle tuotteelle saattaa käynnistää innovaatioprosessin – tätä kutsutaan markkinavetoiseksi innovaatioprosessiksi (market pull). (Tidd et al. 2005, s. 75).

Varhaisten mallien heikkoutena on interaktiivisen tiedonkulun puute. Käytännössä innovaatioprosessi vaatii jatkuvaa keskustelua eri toimintojen välillä. Innovaatioprosessi voi lähteä markkinoiden tarpeesta tai teknologian luomasta mahdollisuudesta, mutta menestyksekkään innovaation luominen edellyttää kummankin innovaatioprosessin lähteen, “työnnön” ja “vedon”, välistä vuorovaikutusta. Tidd esittää havainnollistavan vertauskuvan saksista; jotta saksilla voitaisiin leikata, tarvitaan molempia teriä. (Tidd et al. 2005, s.75)

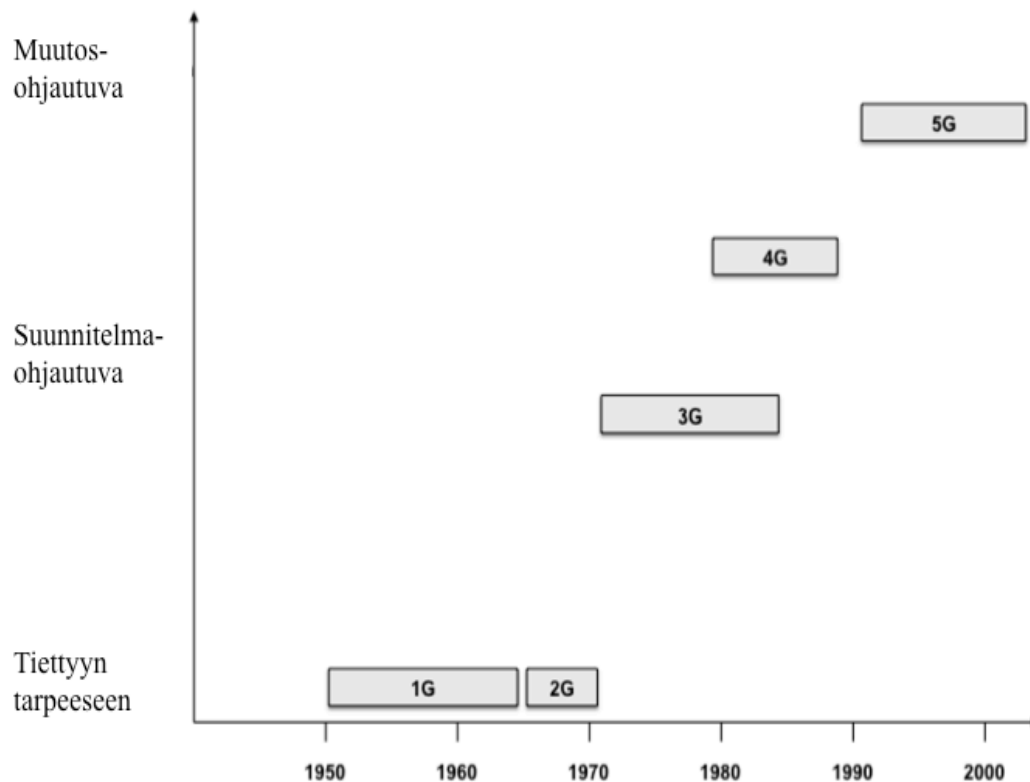
Innovaatioiden on huomattu kehittyvän monimutkaisesti suhteessa aikaan. Lineaarisen innovaatioprosessin mallista on löydetty selkeitä heikkouksia, jotka on lueteltu alla;

- **Sysäykset laukaisevat innovaatioita:** muutos tapahtuu kun ihmiset tai organisaatiot saavuttavat mahdollisuuden tai tyytymättömyyden kynnyksen.
- **Ideat lisääntyvät:** lähdettyään yhdestä ideasta innovaatioprosessi synnyttää uusia ideoita, jotka synnyttävät eri suuntiin lähteviä kehityskulkuja.
- **Vastoinkäymisiä esiintyy jatkuvasti,** suunnitelmat ovat ylioptimistisia, sitoutuminen projektiin laajenee (commitments escalate) ja virheet saattavat ruokkia uusia virheitä.

- Innovaatioprosessin työryhmä voi muuttua nopeasti organisaation muutosten tai odottamattomien tapahtumien takia.
- **Johdolla on suuri rooli innovaatioprosessin tukemisessa**, mutta myös innovaatioprosessin **kritisoimisessa ja muovaamisessa**.
- Innovaatio edellyttää oppimista, mutta useat innovaatioprosessin tulokset ovat innovaatioprosessin kehittymisen seurausta, jolloin oppiminen saatetaan nähdä “harmaana” alueena innovaatioprosessissa. (Tidd et al. 2005, s. 76)

Innovaatioprosessit ovat kehittyneet yksinkertaisista lineaarisista malleista (1. ja 2. sukupolvi) kohti monimutkaisia ja vuorovaikuttaisia malleja (3.-5. sukupolvi). Yhteistä kaikille malleille on kuitenkin samankaltainen vesiputousmallin pohja. (Tidd et al. 2005, s. 76)

Kuvassa 4 on esitetty innovaatioprosessien evoluutio suhteessa aikaan. Pystysuunnassa kuvataan innovaatioprosessin muutosohjautuvuuden astetta.



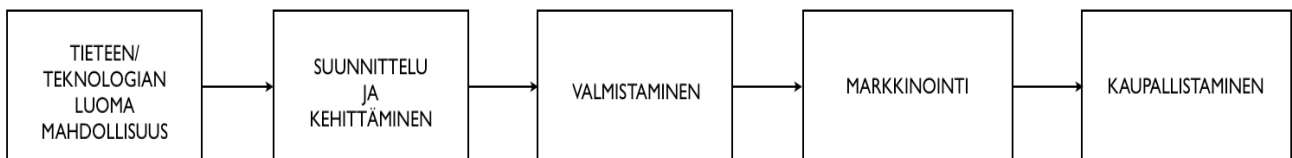
Kuva 4. Innovaatioprosessin evoluutio Rothwellin mukaan. (Rothwell 1994, s. 7)

2.3.1 1G: Teknologiahyönteinen sukupolvi

Toista maailmansotaa seuranneiden kahden vuosikymmenen aikana kehittyneet markkinataloudet kasvoivat ennennäkemättömän nopeasti suurilta osin teollisuuden nopean kasvun ansiosta. Teknologian kehitys mahdollisti uusien alojen kehittymisen ja vanhoilla teollisuuden aloilla nähtiin teknologiajohtoinen tuotantoprosessien ja laadun kehittyminen ja uudistuminen. Tämä kehitys johti työllisyyden nopeaan kasvuun, hyvinvoinnin kehittymiseen ja kysynnän kasvuun erityisesti kodinkoneiden, kuluttajaelektronikan ja autoteollisuuden aloilla – kysyntä saattoi jopa ylittää tuotannon kapasiteetin. (Rothwell 1994, s.7)

Ensimmäisen sukupolven innovaatioprosessit esiintyivät 1950-luvulta 1960-luvun puoleen väliin saakka. Tälle ajanjaksolle oli tyypillistä suuri usko tieteelliseen kehitykseen, teolliseen innovaatioon ja teknologian kaikkivoipaisuuteen yhteiskunnan ongelmien ratkaisijana. Innovaatioprosessi nähtiin lineaarisena prosessina alkaen tieteellisesti löydöstä, käyden läpi tuotekehitysprosessin yrityksen sisällä ja loppuen markkinoille. Teknologiahyönteinen innovaatio-ajattelu lähti siitä oletuksesta, että panostus tutkimukseen ja tuotekehitykseen näkyy suoraan uusina menestyvinä tuotteina yrityksen portfolioissa. (Rothwell 1994, s. 7)

Ensimmäisen sukupolven teorioille yhtenäistä on se, että innovaatioprosessin katsotaan lähtevän teknologian työnnöstä. Prosessi oli lineaarinen ja sai alkunsa tieteellisen tai teknologisen kehityksen luomasta mahdollisuudesta, jolla saatiin uusi tuote “työnnettyä” markkinoille. (Galanakis 2006, s.1223) Ensimmäisen sukupolven teknologiahyönteinen innovaatioprosessi on esitetty kuvassa 5.



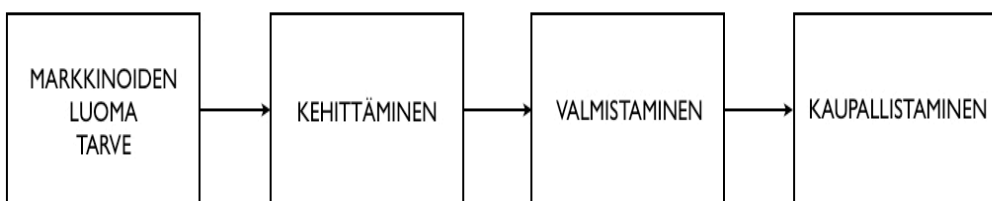
Kuva 5. Teknologiahyönteinen innovaatioprosessi. (Rothwell 1994, s. 7)

2.3.2 2G: Markkinavetoinen sukupolvi

Mentäessä 1960-luvun loppua päin teollisuuden tuotannon kapasiteetti jatkoi kasvamistaan ja yleisen hyvinvoinnin taso pysyi korkeana. Monissa maissa teollisuuden työllisyys pysyi muuttumattomana, mutta teollisuuden tuottavuus lisääntyi nopeaa vauhtia. Yritykset kasvoivat luonnostaan, yritysostojen kautta ja monipuolistuivat nopeaa vauhtia. Uudet tuotteet perustuivat suurimmilta osin olemassa oleviin teknologioihin ja monilla aloilla kysyntä ja tarjonta olivat tasapainossa. Kilpailun kovetessa yritykset panostivat suuresti markkinointiin, taistellessaan markkinaosuudestaan. Innovaatioprosessin pääpaino siirtyi markkinoiden muutoksen johdosta korostamaan kysyntää, teknologian mahdollistaman tarjonnan sijaan. (Rothwell 1994, s. 8)

Markkinavetoinen teoria vallitsi 1960-luvulla ja lähti ajatuksesta, että markkinoiden kysyntä sai aikaan “vedon” tietyn tuotteen saamiseksi markkinoille. Markkinavetoinen teoria oli myös lineaarinen innovaatioprosessin malli. (Galanakis 2006, s.1224)

Tämän markkinavetoisen prosessin pääidea oli, että markkinoiden kysyntä ohjasi tutkimusta ja tuotekehitystä, jonka rooli oli käytännössä reagoida markkinoiden signaaleihin. Markkinavetoinen innovaatioprosessi on esitetty kuvassa 6.



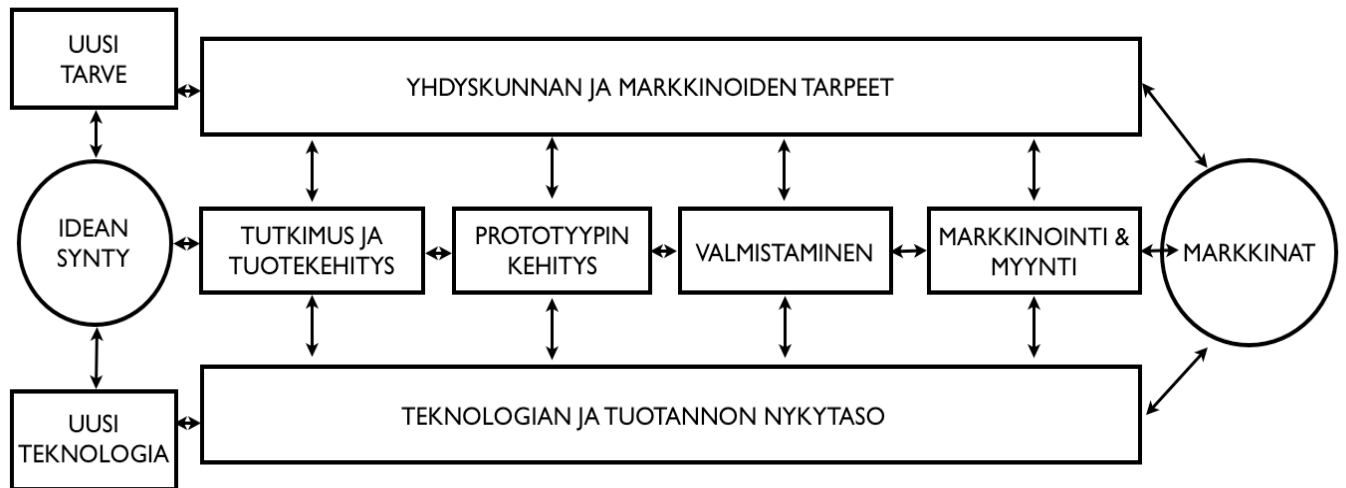
Kuva 6. Markkinavetoinen innovaatioprosessi. (Rothwell 1994, s. 8)

2.3.3 3G: Interaktiivinen sukupolvi

Kahden ison öljykriisin johdosta 1970-luvun länsimainen talous koki samanaikaisesti laman, suurtyöttömyyden sekä korkean inflaation. Tarjonta oli selvästi kysyntää korkeampi ja yritykset joutuivat tekemään suuria panostuksia vakauttaakseen ja rationalisoidakseen toimintaansa. Talouden tilanne johti tarkkaan rahavirran tarkkailuun, siirtäen strategisen pääpainon kustannusten hallintaan ja vähentämiseen. Vuosikymmenen kestänyt yritysten resurssien rajoittuneisuus ohjasi yritykset todella tutkimaan innovaatioprosessia, jotta kaikki mahdollinen "hukka" pystyttäisiin poistamaan. (Rothwell 1994, s.9)

Interaktiivisen innovaatioprosessin teoria vallitsi 1970- luvulla ja 1980-luvun alkupuoliskolla. Teorialle oli ominaista "työntö-veto" – näkemys, jonka mukaan innovaatioprosessi koostuu perättäisistä vaiheista, mutta ei välttämättä ole jatkuva. Teorian mukaan innovaatioprosessi voidaan jakaa useisiin toisistaan erillisiin vaiheisiin ja palautesilmukoihin eri vaiheiden välillä (feed back loops). Yrityksen sisäinen organisaatio, yrityksen ulkoiset kytkökset ja vaikutteet luovat monimutkaisen verkon, joka sitoo yhteen yrityksen omat toiminnot, teknologisen ja tieteellisen yhteisön sekä markkinat. (Galanakis 2006, s.1224) Kolmannen sukupolven teorian mukainen innovaatioprosessin malli on esitetty kuvassa 7.

Innovaatioprosessia tutkittaessa huomattiin, että onnistuminen tai epäonnistuminen pystyttiin hyvin harvoin selittämään vain muutaman päätekijän avulla. Onnistuminen vaati usein enemmän kuin vain muutaman tehtävän hoitamisen erinomaisesti – onnistuminen vaati koko innovaatioprosessin läpi hyvin johdettua ja tasapainotettua tehtävien hoitamista. Huomattiin myös, että onnistuneen innovaatioprosessin "sydäimestä" löytyi aina muutama avainhenkilö, joiden korkeatasoisen osaamisen ja motivaation ansiosta innovaatio oli mahdollista. (Rothwell 1994, s.11)



Kuva 7. Kolmas eli interaktiivinen sukupolvi. (Rothwell 1994, s. 10)

2.3.4 4G: Toiminnot integroiva sukupolvi

Toiminnot integroiva innovaatioprosessin teoria (The Functional Integration Innovation Process Theory) kehittyi kun havaittiin erityisesti Japanin auto- ja elektroniikkateollisuuden käyttävän tiettyntyyppisiä metodeja. Teorian mukaan innovaatioprosessiin osallistuvat yrityksen toiminnot toimivat rinnakkain, eivätkä perättäisinä toimintoina. Teorian ydin on yrityksen eri toimintojen integrointi innovaatioprosessissa siten, että saadaan yhdistettyä eri alojen erityisosaaminen tehokkaasti. Tällä tavoin pystytään innovaatioprosessin kestoa lyhentämään ja uudelleen tekemisen määrää vähentämään eri toimintojen ollessa jatkuvassa vuorovaikutuksessa toistensa suhteen. (Galanakis 2006, s. 1224)

Rothwellin mukaan neljännen sukupolven innovaatioprosessille on olennaista, että toimittajat ja jakelijat ovat alusta alkaen mukana uuden tuotteen kehitysprosessissa ja samalla talon sisäiset funktiot toimivat integroidusti rinnakkain, eivätkä perättäin. Jos täydellinen samanaikainen kehitystyö ei ole mahdollista tai tarpeellista, on osastojen integrointi ja nopea tiedonsiirto osastojen välillä tärkeässä osassa innovaatioprosessia. (Rothwell 1994, s. 12) Neljännen eli toiminnot integroivan innovaatioprosessin malli on esitetty kuvassa 8.



Kuva 8. Neljäs eli toiminnot integroiva sukupolvi. (Rothwell 1994, s.12)

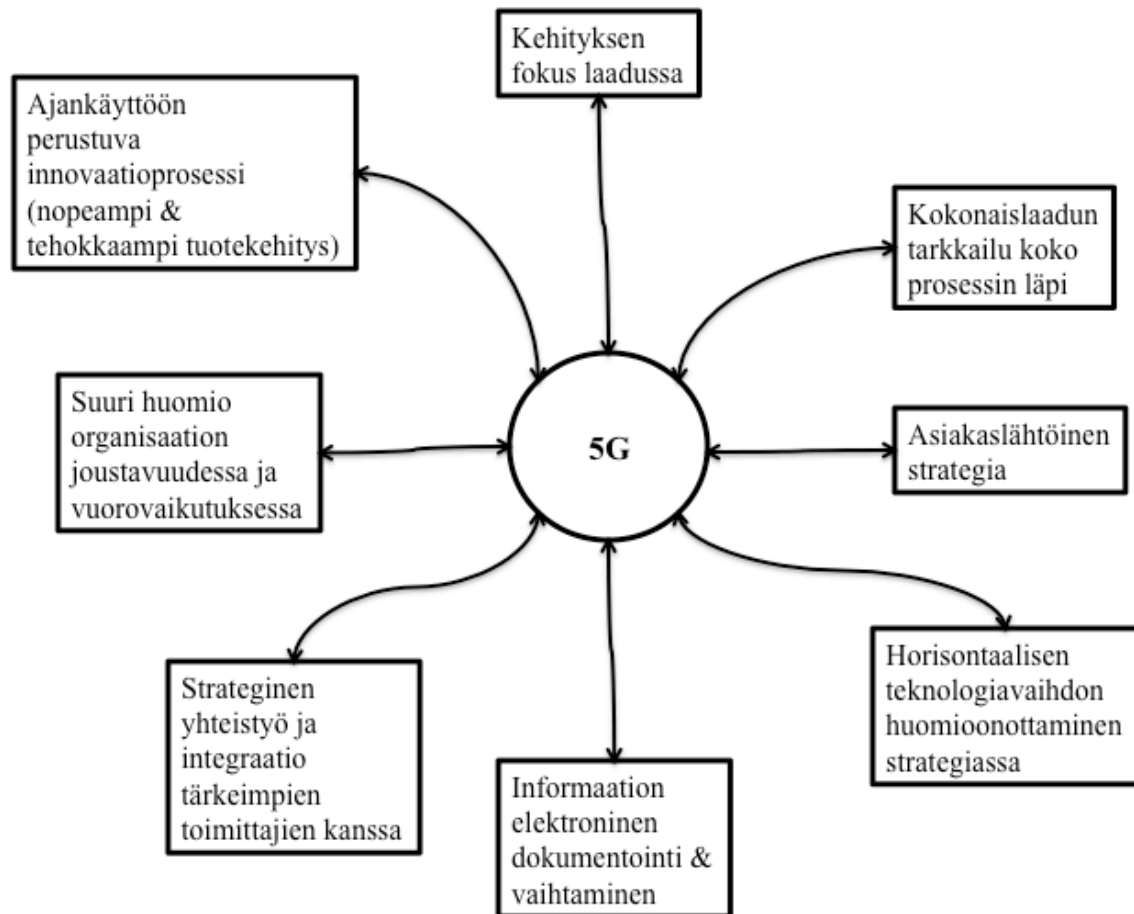
Kuvassa 8 näkyvä esitystapa 4G-innovaatioprosessista keskittyy kahteen sisäiseen pääprosessiin. Näiden pääprosessien ympärillä on 3G-mallissa esitetty ulkoisten toimintojen verkko. (Rothwell 1994, s. 12)

2.3.5 5G: Järjestelmät integroiva verkottunut sukupolvi

Järjestelmät integroivan verkottuneen sukupolven malli (The Systems Integration and Networking Innovation Process Theory) kehittyi neljännen sukupolven mallin pohjalta, painottaen jatkuvan muutoksen tarvetta. Innovaatioprosessissa käytetään teknologiaa suuresti hyödyksi erilaisten mallintamisohjelmien (CAD/CAM) avulla, sekä tekemällä nopeasti prototyyppijä tuotteesta. Mallintamisohjelmilla ja prototyyppien kehittämisen avulla pystytään helpottamaan suunnittelun ja kehityksen vaiheita innovaatioprosessissa. Innovaatioprosessiin kuuluu myös toimittajien, asiakkaiden ja toisten yritysten välisen verkon luonti, jotta pystytään ottamaan hyöty kehittyvistä teknologioista ja ratkaisemaan uusien tuotteiden korkeaan monimutkaisuuteen liittyvät ongelmat. Innovaatioprosessin tehokkuus ja nopeus perustuvat tiedonkulun nopeuteen ja tehokkaaseen käyttöön, sekä jatkuvaan kommunikointiin eri toimintojen välillä koko innovaatioprosessin läpi. (Galanakis 2006, s. 1224)

Viidennen sukupolven radikaalein muutos neljänteen sukupolveen verrattuna on vahvojen elektronisten työkalujen käyttö eli erilaisten mallinnus ja simulaatio-ohjelmien käyttö. Erilaisten prototyyppien teko jo tuotekehityksen alkuvaiheessa, tekniset piirustusohjelmat läpi prosessin ja lopullisen prototyypin/ valmiin tuotteen simulointi 3D-mallinnuksella ovat ominaista 5G-

innovaatioprosessille. (Rothwell 1994, s.22) Viidennen sukupolven innovaatioprosessin mallin tärkeimmät strategiset pääkohdat on esitetty kuvassa 9.



Kuva 9. 5G:n tärkeimmät strategiset pääkohdat. (Galanakis 2006, s. 1224)

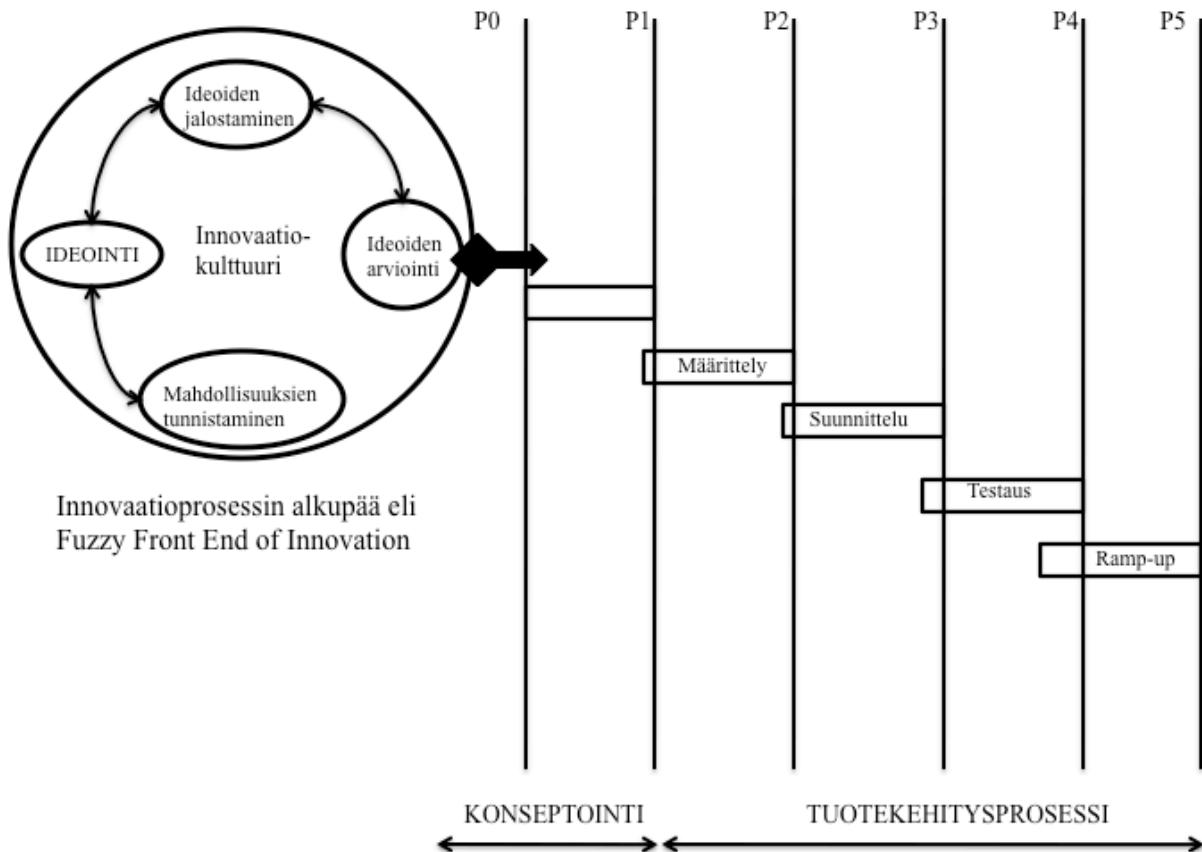
3 Nykyaikaisen innovaatioprosessin ongelmat

Innovaatioprosessi kattaa laajemman kokonaisuuden kuin perinteiden tuotekehitysprosessi. Varsinaisen tuotekehitysosuuden lisäksi innovaatioprosessiin kuuluu mahdollinen tutkimusosuus, sekä innovaatioprosessin alkupää. (Apilo et al. 2007, s. 132)

Innovaatiota ja innovaatioprosessia käsittelevä kirjallisuus käsittelee innovaatioprosessin ongelmia lähinnä alkupään ja tutkimusosuuden osalta, joten voidaan olettaa, että innovaatioprosessille ominaiset ongelmat löytyvät prosessin alusta. Innovaatioprosessi sisältää myös tuotekehitysosuuden, joten on perusteltua tutkia myös suurimpia tuotekehitysprosessin ongelmia.

Kuten aiemmin todettiin, on innovaatioprosessi kehittynyt yksinkertaisista lineaarisista malleista monimutkaisiin ja vuorovaikutteisiin verkostomalleihin. Lineaaristen mallien heikkoutena oli interaktiivisen tiedonkulun puute ja prosessin jäykkyys. Vuorovaikutteisemmat mallit ovat paikanneet lineaaristen mallien heikkouksia, mutta uudeksi ongelmaksi on noussut innovaatioprosessin alkupäässä vallitseva epäselvyyden tila.

Nykyaikainen innovaatioprosessi voidaan jakaa kolmeen osaan; alkupää, tuotekehitys ja kaupallistaminen. (Kahn 2005, s. 82) Alkupään ja tuotekehitysosuuden välissä voidaan vielä nähdä yksi välivaihe: konseptointi. Tämä nykyaikainen näkemys, sisältäen konseptoinnin on esitetty kuvassa 10.



Kuva 10. Innovaatioprosessi sisältäen alkupään, konseptoinnin ja tuotekehitysprosessin. (Apilo et al. 2006, s. 43)

3.1 Alkupää ja konseptointi

Innovaatioprosessin alkupäässä yritys muodostaa käsityksen teknologioiden, markkinoiden ja asiakkaiden tarpeiden kehittymisestä tulevaisuudessa. Alkupäässä valitaan jatkokehittävät ideat, jotka takaavat yrityksen menestymisen. Innovaatioprosessin loppupäätä lähestyttäessä, vaikutusmahdollisuudet prosessin kulkuun vähenevät ja kallistuvat huomattavasti. Innovaatioprosessin alkupää tarjoaakin suurimmat haasteet, mutta myös mahdollisuudet. (Apilo et al. 2007, s. 132)

Englanninkielinen nimitys innovaatioprosessin alkupäälle on ”fuzzy front-end”. Innovaatioprosessin alkupäätä pidetäänkin usein kaaosmaisena tai sumeana. Suurin ero alkupään ja muun innovaatioprosessin välillä on se, että alkupää on jatkuvaa toimintaa ja loppupää projekteja. Alkupäätä ei voida tuotekehitysosuuden tavoin määritellä tiukkaan prosessimuotoon, mutta

alkupään tunnistetut tehtävät voidaan luokitella seuraavasti; mahdollisuuksien tunnistaminen, ideointi, ideoiden kehittäminen ja ideoiden arvioiminen. (Apilo et al. 2007, s. 134)

Alkupään ongelmista esiin nousee asiakastarpeen ymmärtäminen. Organisaation tulisi pyrkiä antamaan hyvät edellytykset mahdollisuuksien tunnistamiseen, erityisesti teknologian puolella – sekä erilaisia työkaluja asiakastarpeiden ymmärtämiseen. Usein asiakas ei osaa sanoa, mitä tarvitsee. Asiakkaan on usein vaikea artikuloida tuotetta tai ratkaisua, jota ei ole vielä markkinoilla, mutta ongelman asiakas osaa usein kuvata, jos häntä autetaan. (Apilo et al. 2007, s. 135)

Alkupään tavoitteet ovat usein ristiriidassa. Alkupäätä hallitsee usein teknologia- ja markkinalähtöisyyden vastakkainasettelu. Runsas tutkimustieto korostaa menestyneiden innovaatioiden takana olevan asiakkaiden tarpeiden ymmärtäminen. Vaatimus nopeaan tuotekehitykseen ja konseptointivaiheessa tehdyt ratkaisut niin hinnan kuin asiakastarpeiden toteutumisen kannalta asettavat ristiriitaisia tavoitteita innovaatioprosessin alkupäälle. (Apilo et al. 2006, s. 45)

Konseptointivaiheen suurimmiksi ongelmiksi nousevat; 1) asiakkaiden tarpeiden ja prosessien tunnistaminen, ymmärtäminen ja muokkaaminen tuotevaatimuksiksi, 2) konseptointivaiheen resurssointi eli oikeiden osaamisten kytkeminen konseptointivaiheeseen, sekä 3) konseptien liiketoiminnallisen hyödyn arviointi ja liiketoimintatiedon hyödyntäminen. (Apilo et al. 2006, s. 47)

Innovaatioprosessiin liittyy monia ongelmia; alkupään sekavuus, “varaslähdöt”, kierrättäminen eri toimintojen välillä ja umpikujat. Innovaatioprosessin ongelmankenttää on yritetty kuvata rautatiemallilla, jossa innovaatioprosessi on kuin rautatieverkosto, sisältäen pysäkkejä, sivuraiteita ja jopa silmukoita edellisille pysäkeille. (Tidd et al. 2005, s. 76)

3.2 Tuoteprosessi

Innovaatioprosessi etenee alkupäästä konseptointiin ja siitä tuoteprosessiin. Tuoteprosessilla tai uuden tuotteen kehitysprosessilla tarkoitetaan yleensä ajanjaksoa, joka useimmilla yrityksillä alkaa konseptoinnista ja päättyy volyymivalmistukseen. Tuoteprosessi toteutetaan yleensä projekteina. Tuoteprosessin tavoiteltavin ominaisuus on nopeus, johon päästään poikkifunktionaalisilla tiimeillä ja yhtäaikaisella suunnittelulla. (Apilo et al. 2007, s. 158)

Useat (suomalaiset) yritykset ovat alkaneet kuvata tuoteprosessia prosessikuvauksilla, mutta ongelmaksi on havaittu tuoteprosessin vakiinnuttamisvaiheessa dokumenttien määrän ja kattavuuden, sekä katselmusten turhan suuren määrän. Prosessit suunnitellaan haastavimpien projektien kuvauksiksi, mutta merkittävä osa kehitysprojekteista voitaisiin viedä läpi pienemmälläkin byrokratialla. (Apilo et al. 2007, s. 160)

Voitto- projektin loppuraportin pohjalta Apilo & Taskinen (Apilo et al, 2006) esittävät, että on hämmästyttävää, että uusia tuotteita saadaan kehitettyä nykyisellä aikataululla – paljon väärinymmärryksiä, värien piirustus- ja kehitysversioiden ja kommunikaatiokatkosten aiheuttamaa ylimääräistä työtä esiintyy jo yhden yrityksen sisällä. Asioiden hallinta ei ainakaan muutu helpommaksi, kun tuotekehitys siirtyy yhä enemmän verkostoon. (Apilo et al. 2007, s. 163)

Myös realistisen arvion tekeminen tuoteprosessin laajuudesta ja aikataulusta tuottaa ongelmia. Jos tuotteen lanseerauspäivä ei riipu jostain alan tärkeästä messusta tai sesongista, on turhaa määritellä tarkkaa päivämäärää tuotteen lanseeraukselle. (Apilo et al. 2007, s. 161) Ongelmana alun kiinteisiin määrittelyihin ja tuoteprosessin läpivientiin liittyen nähdään tuoteprosessin liian ”putkimainen” läpivienti. Vaikka käytettäisiin tarkastuspisteitä tuoteprosessin eri vaiheissa, ovat tarkastuspisteet harvoin ”jatketaan/ei jatketa”-pisteitä. Tuoteprosessin lopettamista keskeneräisenä otetaan harvoin huomioon, vaikka se olisi paras mahdollinen vaihtoehto. (Apilo et al. 2007, s. 165) Kun nämä ongelmat yhdistetään vaikeuteen ymmärtää ja muuttaa asiakkaan tarpeet tuoteominaisuuksiksi, voidaan olettaa, että innovaatioprosessin alkupäähän liittyy samankaltaisia ongelmia kiinteiden määrittelyjen suhteen kuin järjestelmäkehitysprosessissa.

Yhtenä haasteena nähdään tiedon ja osaamisen siirtäminen poikkifunktionaalisten tiimien välillä. Erilaisten jälleenarviointitilaisuuksien järjestäminen ja projektien loppuraporttien tutkiminen voi auttaa, mutta menetelmien ja dokumenttien käyttö ei auta, ellei tiedon ja uuden osaamisen välittämisen tärkeyttä painoteta. (Apilo et al. 2007, s. 164)

Yhteenvetona voidaankin todeta tuoteprosessin merkittävimpien ongelmien liittyvän prosessin jäykkyyteen ja tiedonvälitykseen. Ongelman muodostavat liika dokumentaatio ja työläät katselmuksiset sekä kommunikaatiokatkokset ja vaikeudet tiedon sekä osaamisen siirtämisessä.

4 Järjestelmäkehitysprosessi

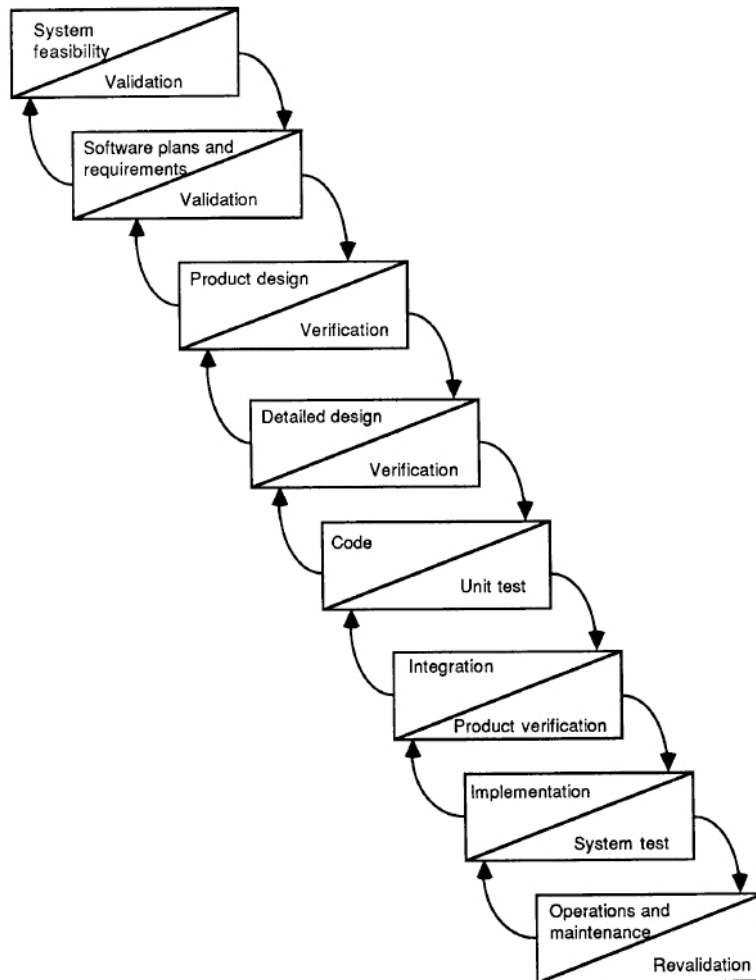
Järjestelmäkehitysprosessilla tarkoitetaan prosessimallia, jota käytetään ohjelmistokehityksen eri vaiheiden ohjaamisessa sovellussuunnittelussa. Termi järjestelmäkehitysprosessi on syntynyt Pohjois-Atlantin puolustusliiton, NATO:n, vuonna 1968 koolle kutsuman konferenssin nimestä ja sen keskeinen sisältö on alkujaan konferenssissa työstetyn ideaaliprosessin mukainen. Prosessi nimettiin tahallisen ristiriitaisesti järjestelmäkehitykseksi (software engineering), jolla haluttiin viitata perinteisiin insinööritieteisiin ja vielä 60-luvulla kehittymättömään ja strukturoimattomaan kasvavaan tieteeseen. (NATO Science Committee 1969, s. 8)

Ohjelmistokehityksessä perinteiset tilamallin mukaiset prosessit pyrkivät paloittelemaan käsiteltävän ongelman niin pieniin osiin kuin mahdollista. Äärimmilleen vietynä tämä tarkoittaa yksittäisten luokan sisällä olevien metodien tai aliohjelmassa olevien ohjelmalohkojen syötteiden ja tuotteiden tarkkaa määrittelyä ennen toteutuksen aloittamista. Periaate on Taylorismina tunnettu liikkeenjohdon oppi, jota Fordin liukuhihnalla toteutettiin 1915, kun Fordin liukuhihnalla 7000 asentajaa rakensivat T-Fordia toistamalla omaa yksittäistä rutiiniaan kerrasta toiseen ja autoyksilöstä toiseen. Fordilla oli laumoittain insinöörejä suunnittelemassa ja parantamassa asentajien työtehtäviä, valvomassa niiden suoritusta ja opastamassa niiden tekemisessä, jotta autot saataisiin hihnan läpi mahdollisimman nopeassa ajassa. (Womack, Jones & Roos 1991, s. 31)

4.1 Vesiputousmallin historia ja kehitys

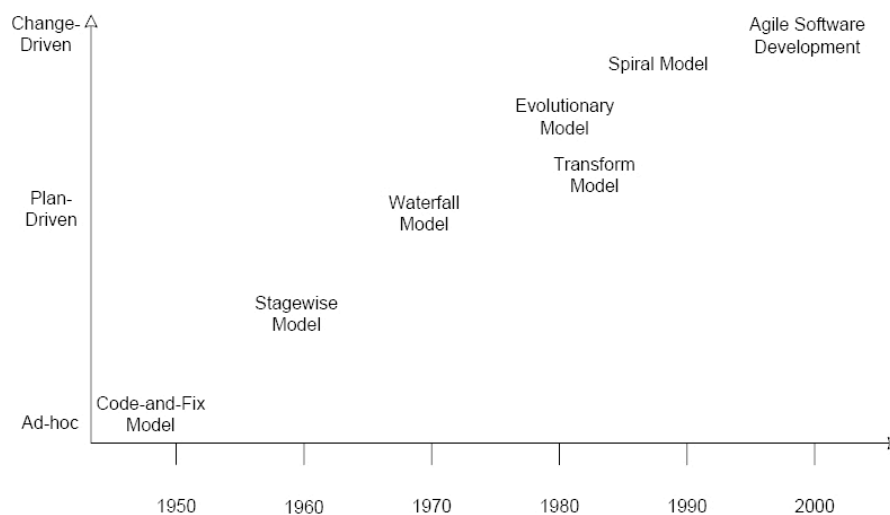
Ensimmäiset ohjelmointiprosessit olivat äärimmäisen yksinkertaisia: ensin koodattiin ja sitten testattiin ja korjattiin. Vaatimusten täyttymistä, arkkitehtuuria ja koodin ylläpidettävyyttä mietittiin vasta varsinaisen ohjelmoinnin jälkeen. Ohjelmien monimutkaistuessa myös ongelmat muuttuivat entistä pahemmiksi ja 50-luvun puolessa välissä esiteltiin ensimmäinen vaiheittainen ohjelmistokehitysmalli eli niin kutsuttu tilamalli, missä määrittely ja suunnittelu oli sijoitettu ennen ohjelmointia ja testaus ja evaluointi sen jälkeen. Vesiputousmalli esiteltiin vuonna 1970 (Royce 1970) ja siinä otettiin suurilta osin vaikutteita tilamallista. Vesiputousmalli tarjosi kuitenkin kaksi uutta ratkaisua tilamallissa havaittuihin ongelmiin: ensinnäkin siinä otettiin huomioon peräkkäisten vaiheitten välisen palautteen tarve ja toiseksi siinä esiteltiin prototyypin rakentaminen rinnakkain määrittely- ja suunnitteluvaiheiden aikana. Mielenkiintoista on, että jo vesiputousmallin esittelyssä Royce kiinnitti huomiota asiakkaan sitouttamiseen ohjelmistotuotantoprosessin lopputuloksiin, liittämällä asiakas mukaan prosessin eri vaiheissa suoritettaviin katselmointeihin (Royce 1970, s.

335). Roycen mukaan asiakkaan on oltava prosessissa mukana formaalisti, syvällisesti ja jatkuvasti (Royce 1970, s. 337). (Boehm 1988, s. 62-63)



Kuva 11. Vesiputousmalli (Boehm 1988, s. 62)

Vuosikymmenten saatossa vesiputousmallia on kehitetty edelleen ja Salo (2006) kuvaakin ohjelmistokehitysprosessin kehittyneen monimutkaisuuden ja ongelmanratkaisun hallitsemiseksi alkujaan ad-hoc mallista tilamallin ja vesiputousmallin kautta muutospohjaiseen malliin ajassa seuraavasti:



Kuva 12. Ohjelmistokehitysmallien evoluutio (Salo, 2006, s.18)

Pyrkimys ohjelmistokehitysprosessin aikana tapahtuvien muutosten hallintaan johti Salon (2006, s. 21) mukaan iteratiivisen ja inkrementaalisen mallin käyttöönottoon. Boehm (1988, s. 61-65) kuvaa artikkelissaan eri ohjelmistokehitysmallit ja niiden ongelmat lyhyesti esitellessään spiraalimallin josta tulikin Larmanin ja Basilin (2003, s. 51) mukaan iteratiivisen ja inkrementaalisen ohjelmistokehityksen maamerkki. Larmanin ja Basilin (2003, s. 50) mukaan NASA käytti avaruussukkulaohjelmassaan vuosina 1977-1980 iteratiivista ja inkrementaalista kehitystä vesiputousmallin sijaan pystyäkseen mukautumaan sukulaohjelman toteutusvaiheen jatkuviin muutoksiin. Muita esimerkkejä inkrementaalisen ja iteratiivisen kehityksen käytöstä Larman ja Baisili (2003) esittelevät artikkelissaan jo 50-luvulta 90-luvulle ja heidän mukaansa malli onkin ollut läsnä ohjelmistokehityksessä jo alusta alkaen.

4.2 Vesiputousmallin ongelmat

Keskeisiä ongelmia monessa laatujärjestelmässä ovat mm. niiden oletus, että suunnitelma tai kehitysprosessi voidaan toteuttaa kerralla oikein ja tästä oletuksesta aiheutuneet jäykät ja työläät määrittely-, suunnittelu- ja auditointiprosessit (Poppendieck ja Poppendieck 2003, Preface XVIII). Boehm (1988, s. 63) väittää vesiputousmallin suurimman ongelman olevan sen määrittely- ja suunnitteluvaiheiden vaatima raskas dokumentaatio. Schwabin mukaan vesiputousmallin suurin ongelma on kyvyttömyys reagoida yllättäviin osatuloksiin kesken lineaarisen prosessin (Schwaber 1995, s. 5).

Toinen merkittävä ongelma, mikä liittyy moniin prosesseihin, on niiden johtamisjärjestelmä. Päätöksenteko keskitetään organisaatiossa harvoille, yleensä hierarkiassa korkealla sijaitsevalle

auktoriteetille (Poppendieck et al. 2003, Introduction XXIII), jolla ei kuitenkaan ole enää käytännössä ongelmien ratkaisemiseen tarvittavia tietoja käytettävissään.

Voidaankin päätellä, kuten Salo (2006) kertoo, että perinteisten suunnitelmiin perustuvien menetelmien suurin ongelma on niiden alkuvaiheiden pyrkimys varmuuteen epävarmuuden hallitsemiseksi. Alkuvaiheen laaja määrittely ja kiinnitetty toteutuslaajuus johtavat kiinteiden resurssien, kustannusten ja aikataulun arviointiin. Sekä asiakkaalla että toimittajalla on yhteinen houkutus epävarmuuden hallitsemiseen kiinteiden määrittelyjen kautta. (Salo, 2006, s. 20)

5 Ketterät menetelmät

Abrahamssonin, et.al mukaan (Abrahamsson, Salo, Ronkainen & Warsta. 2002, s.17, s. 98) ohjelmistokehitysmenetelmä on ketterä, kun se täyttää neljä vaatimusta. Ketterä ohjelmistokehitys on:

- inkrementaalista (pienissä erissä, tiheä julkaisutahti)
- yhteistoiminnallista (asiakas ja kehitystiimi työskentelevät ja kommunikoivat jatkuvasti toistensa kanssa)
- suoraviivaista (menetelmä on helppo omaksua ja muokata sekä hyvin dokumentoitu) ja
- mukautuvaa (viimehetken muutoksia voidaan ottaa ohjelmistoon mukaan).

Cockburnin (2002) mukaan ketterä (Agile) ohjelmistokehitys syntyi helmikuussa vuonna 2001, kun joukko ohjelmistokehityksen kevyen sarjan menetelmien kannattajia kokoontui yhteen tutkiakseen menetelmiensä yhteisiä piirteitä. Hänen mukaansa oli suorastaan yllättävää, että paikalle saapuneet 17 individualistista asiantuntijaa pystyivät olemaan yhtä mieltä ylipäättään mistään. Kuitenkin kokoontumisen lopputuloksena syntyi Agile Alliance, ketterien ohjelmistokehitysmenetelmien ydinryhmä, joka julkaisi manifestin paljastaakseen parempia tapoja kehittää ohjelmistoja käyttämällä ja opettamalla muita käyttämään jo olemassa olevia tapoja. Manifesti julkaistiin seuraavien neljän vastakohtan avulla:

1. yksilöt ja vuorovaikutus vastaan prosessit ja työkalut
2. toimiva ohjelmisto vastaan täydellinen dokumentaatio
3. asiakasyhteistyö vastaan sopimusneuvottelut
4. muutosherkkyys vastaan suunnitelman noudattaminen. (Cockburn 2002, s. 215-218)

Ketterissä menetelmissä siis lähestytään perinteisen ohjelmistokehityksen ongelmia toisesta lähtökohdasta - lopullista tuotetta ei pyritäkään saamaan kerralla valmiiksi, vaan asiakkaan kanssa määritelty tuote kootaan vaiheittain ja vaatimuksista asiakkaalle tärkeimmät toteutetaan ensin. Näin saadaan asiakkaalle ensimmäinen valmis versio tuotteesta mahdollisimman nopeasti ja tällä tavoin pyritään tukemaan asiakkaan kilpailuasemaa markkinoilla ja ollaan valmiita tekemään mahdollisesti muuttuneen tilanteen edellyttämiä muutoksia tuotteeseen. (Poppendieck et al. 2003, s. 70)

Ketterien menetelmien käyttö ei kuitenkaan tarkoita sitä, että systemaattista prosessia ei enää olisi ollenkaan tai kukaan ei johtaisi tekemistä – päinvastoin, Poppendieckin et al. (2003, s. 109), mukaan ohjelmistokehityksen tulee olla kurinalaista jotta työ etenee vaivattomasti, nopeasti ja tuottavasti. Tällä pyritään toisaalta estämään työhönsä erittäin sitoutuneiden työntekijöiden jatkuva ja usein vapaaehtoinen ylikuormittaminen (Poppendieck et al. 2003, s. 110) ja toisaalta tukemaan asiakkaan kanssa priorisoitujen tehtävien toteuttamista (Poppendieck et al. 2003, s. 13).

Ketterissä menetelmissä toteutuu useimpien muiden menetelmien tavoin tarve mitata suoriutumista annetuista tehtävistä. Perinteisistä mittareista poiketen, yleensä ei mitata alkuperäisen määrittelyn, budjetin tai aikataulun pitävyyttä (Poppendieck et al. 2003, s. 25), vaan ennemminkin tiimin jäsenten iteraatioon ottamia ominaisuuksia (Poppendieck et al. 2003, s. 115) ja toteutettujen ominaisuuksien välistä suhdetta (Boehm & Turner 2005, s. 6).

Lyhyesti kiteytettynä ketterien menetelmien keskeinen ero perinteisiin suunnitelmaorientoituneisiin menetelmiin on niiden pyrkimys laajempaan kommunikaatioon ja palautteeseen pienissä osissa asiakkaalle julkaistujen kokonaisuuksien kautta. Seuraavassa tarkastellaan muutamien ketterien ohjelmistokehitysmenetelmien ominaispiirteitä iteratiivisen tai inkrementaalisen lähestymistavan lisäksi.

5.1 Scrum

Scrumin ideologian ajatellaan yleisesti syntyneen Takeuchin ja Nonakan vertauskuvasta artikkelissa *The New New Product Development Game* (1986, s. 2-7), missä he käsittelivät rugby pelin tapaa liikuttaa palloa pelaajien kesken tiiviissä muodostelmassa (engl. scrum) ja hyvin projektissa suoriutuvaa tiimiä keskenään. Myöhemmin DeGrace ja Stahl käyttivät tästä lähestymistavasta nimitystä scrum kirjassaan *Wicked Problems, Righteous Solutions* (1990, 179s.) ja 90-luvun puolessa välissä scrum mallia työstivät Ken Schwaber ja Jeff Sutherland (Cockburn, 2002, xxi).

Schwaberin (1995, s.8) mukaan kehittämisorganisaatio tarvitsee erityisen joustavan työskentelytavan pystyäkseen mukautumaan monimutkaisen ohjelmiston alati muuttuviin vaatimuksiin. Scrum koostuu kolmesta päävaiheesta: suunnittelu (pre-game), toteutus (game) ja lopetus (post-game) ja eri vaiheiden läpi edetään lineaarisesti. Toteutusvaihe jaetaan kuitenkin 1-4 viikon mittaisiin toteutusjaksoihin joita kutsutaan nimellä sprint, missä asiakkaan vaatimuksista toteutetaan valmiit ominaisuudet. Yhdessä ohjelmistokokonaisuudessa voi olla useita sprinttejä. Koko projektin vaatimukset kerätään product backlog –nimiseen varastoon, mistä niitä nostetaan kuhunkin sprinttiin toteutettavaksi. Ajatuksena on, että tekijä itse ottaa tehtäväkseen sen, minkä katsoo pystyvänsä sprintin aikana tekemään. (Schwaber 1995, s.8-14)

Jokaisen sprintin alussa on pidempi suunnittelujakso missä product backlogista valitaan yhdessä asiakkaan kanssa sprintissä toteutettavat osat. Jokaisen päivän alussa on lyhyt suunnittelupalaveri, missä osallistujat sopivat tehtävänsä ja kertovat mahdollisista eteensä tulleista ongelmista. Menetelmässä on erityinen Scrum Master –rooli, jonka omaava, yleensä kokeneempi asiantuntija, tukee ensisijaisesti tiimiä ongelmanratkaisussa. (Abrahamsson et al. 2002, s.33-34)

5.2 Extreme programming

Extreme programming (XP) on eittämättä tunnetuin ohjelmistokehityksen ketterä menetelmä. Menetelmän kehitti Kent Beck, Ron Jeffries ja Martin Fowler vuonna 1996, työskennellessään Daimler-Chrysler yhtiössä ongelmiin ajautuneen projektin nimeltä C3 ohjelmiston kanssa (Beck 1999a, s.75). Menetelmä nojaa neljään perusarvoon: kommunikointiin, yksinkertaisuuteen, palautteeseen ja rohkeuteen (Cockburn 2002, 166s.). Kuten Beck (1999a) artikkelinsa alussa kirjoittaa, XP ei oikeastaan sisällä uusia työmenetelmiä, vaan ennemminkin valjastaa ne käyttöön äärimmäisyyksiin viedyllä tavalla – siitä nimikin on saanut alkunsa (Beck 1999b, s.7).

XP:n keskeinen osa on niin sanottu ”Planning Game” missä asiakas aikatauluttaa ohjelmiston ominaisuudet erillisiin julkaisuihin ohjelmoijilta saamiensa työmääräarvioiden avulla. Vaatimukset kirjataan ja kerrotaan käyttäjätarinoina jotta niiden asiakkaalle tuottama arvo ja sisältö pystytään kuvaamaan molempien asiakkaan ja ohjelmoijan ymmärtävällä tavalla. Vaiheen keskeinen tavoite on jakaa asiakkaan kanssa kaikkein keskeisimmät vaatimukset erillisiin tuotejulkaisuihin niiden tärkeysjärjestyksessä. (Beck 1999a, s.70-77)

Toinen merkittävä ero perinteiseen työskentelytapaan on parityöskentelyn käyttö. XP:ssä käytetään menetelmää nimeltä parityöskentely, missä käytännössä yhden päätteen ääressä työskentelee kaksi

asiantuntijaa, jotka vuorotellen ohjelmoivat ja etsivät esimerkiksi tietoa ongelman ratkaisemiseksi. Rooleja ja pareja voidaan vaihtaa vaikka päivittäin jotta osaaminen ja tietämys ohjelmistosta siirtyvät kaikille tiimin jäsenille. (Beck 1999b, s.51)

Kolmas keskeinen menetelmä on testitapausten kirjoittaminen ennen varsinaista toteutusta. Menetelmää nimitetään yleisesti Test Driven Developmentiksi ja Larman (Larman et al. 2003, s.47) viittaakin sen syntyneen alun perin NASA:n avaruusohjelmien aikana -60 luvulla. Beckin (1999b) mukaan TDD:n avulla ohjelman laatu voidaan varmistaa sekä pitkällä että lyhyellä aikavälillä. Pitkällä aikavälillä testitapaukset varmistavat ohjelmiston ylläpidettävyyden paljon perinteistä ylläpitoa pidemmän ajan ja lyhyellä aikavälillä valmiit testitapaukset toimivat ohjelmoijan vakuutuksena siitä, että tehdyt muutokset eivät ole rikkoneet jo olemassa olevia osia ohjelmistosta. (Beck 1999b, s.44)

XP:ssä ohjelmoijat kehittävät ja testaavat asiakkaan kanssa valitut ominaisuudet erittäin lyhyissä, jopa vain 15 minuuttia kestävässä osissa joiden jälkeen koko ohjelmisto testataan etukäteen kirjoitettujen yksikkötestien avulla. Lyhyet integrointiosat muodostavat päivän tai kahden mittaisia toteutusjaksoja jotka taas kootaan kahden – neljän viikon mittaisiksi suunnittelu ja toteutus jaksoiksi. (Cockburn 2002, s.166-168)

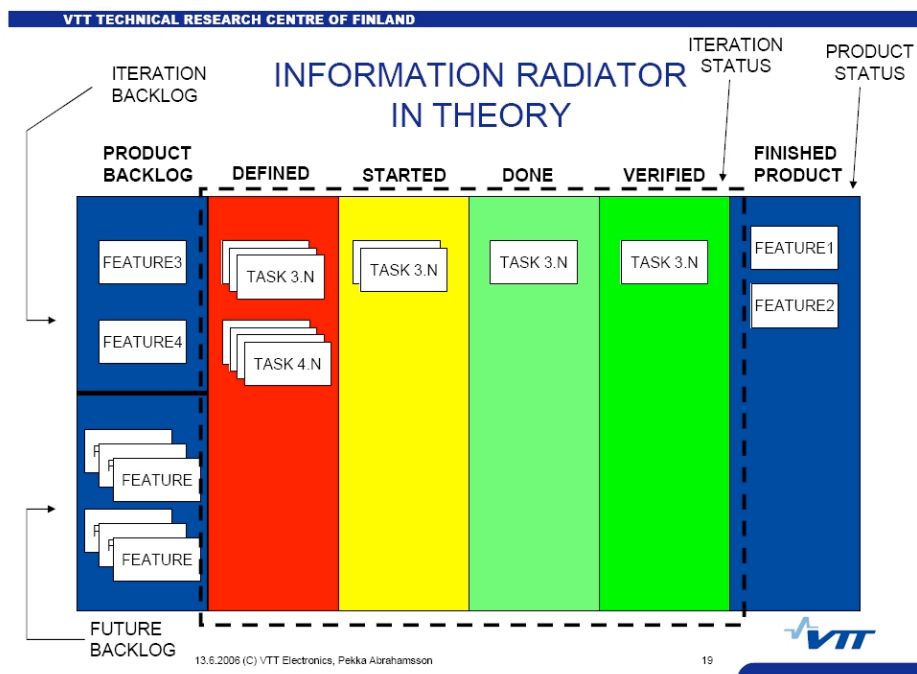
5.3 Mobile-D

Mobile-D on VTT:n kehittämä ketterien menetelmien sovellus erityisesti mobiilisovellusten tuottamiseksi. Menetelmän keskeinen malli on erittäin lyhyeksi puristettu Scrumin sprintti, vain yhden päivän mittainen toteutusjakso nimeltä daily heartbeat. Yksittäiset päivät koostetaan kahden viikon mittaisiksi core –jaksoiksi, joissa koko varsinainen suunnittelu -> toteutus -> julkaisu ketju viedään läpi. Isompi toteutus voi sisältää useita core –jaksoja. Jokaisen päivän työt aloitetaan yhteisellä kokoontumisella, missä lyhyesti kerrataan kunkin edellisen päivän aikaansaannokset ja suunnitellaan alkavan päivän työt. (Abrahamsson, Hanhineva, Hulkko, Ihme, Jäälinoja, Korkala, Koskela, Kyllönen & Salo 2004, s.175)

Mobile-D:n yhteydessä on sovellettu myös Cockburnin (2002, s.90) kuvaamaa ohjelmistokehityksen keskittämistä yhteen tilaan, josta Cockburn käyttää nimeä ”caves and common”. Mobile-D:n yhteydessä siitä on käytetty nimitystä ”War room” (Abrahamsson 2006, s.10). Projektin ajaksi ohjelmistokehitys keskitetään yhteen tilaan, war roomiin, missä on käsillä

kaikki projektin materiaali ja toteutukseen osallistuvat henkilöt. Työskentely samassa tilassa parantaa Cockburnin mukaan osmoottista tiedonvälitystä. (Cockburn 2002, s. 90)

Kolmas implementoitu menetelmä on Cockburnin (2002, s.84) kuvaama tekeillä olevan projektin vaatimusten ja valmiiden ominaisuuksien seurantaan kehitetty ”Information Radiator”. Se on seinälle piirretty lista, kartta tai prosessikuvaus, mistä yhdellä silmäyksellä pystyy seuraamaan projektin tilaa. Mobile-D sovelluksessa kartan on tarkoitus kuvata yhden core-jakson etenemistä ja taustalla on Scrumistakin tuttu product backlog, jonka sisältämät vaatimukset vievät jakson aikana prosessin eri vaiheiden läpi valmiiksi ominaisuuksiksi. Cockburnin (2002, s.85) mukaan yksittäistä vaatimusta kuvaamaan voidaan käyttää yksinkertaisesti Post-it lappuja tai jotain muuta helposti nähtävillä olevaa mediaa, mutta ei mielellään html -sivua. Prosessin eri vaiheita varten kartalle on etukäteen piirretty vastaavat alueet, jotka on ao. kuvassa myös väritetty havainnollisuuden lisäämiseksi. (Abrahamsson 2006, s.19-20)



Kuva 13. Information Radiator (Abrahamsson 2006, s.19)

5.4 LEAN

Lean Development (joskus myös Lean Production) on alun perin Toyotan insinöörin Taiichi Ohnon -40 luvulla kehittämä menetelmä, joka on sittemmin kopioitu laajalti autoteollisuuden käyttöön (Poppendieck et al. 2003, xxiii). Menetelmän ydin on keskittyminen prosessin arvoa tuottamattomiin osiin, jätteisiin tai hukkaan (eng. waste), ja niiden poistamiseen prosessista.

Menetelmään on myöhemmin laajennettu 6 muulla periaatteella, jolloin tullaan nykyisiin seitsemään Lean periaatteeseen (Poppendieck et al. 2003, 66s.). Tässä käsitellään erityisesti ohjelmistokehityksen tarpeisiin räätälöityä LEAN periaatteiden sovellusta, koska periaatteelliset erot kehittämistoiminnan ja ohjelmistoliiketoiminnan välillä ovat pienet verrattuna eroihin perinteiseen tuotannolliseen toimintaan. Seuraavassa periaatteet on esitelty englanniksi ja suluissa vapaasti suomennettuina:

- eliminate waste (poista hukka, jäte)
- amplify learning (vahvista oppimista, varsinkin vaikeissa ongelmissa)
- decide as late as possible (päätä niin myöhään kuin mahdollista)
- deliver as fast as possible (toteuta niin nopeasti kuin mahdollista)
- empower the team (vahvista tiimiä, anna sen toimia täysillä)
- build integrity in (varusta tuotteesi idealla, asiakkaan ajatuksella) ja
- see the whole (näe kokonaisuus, älä osaoptimoi). (Poppendieck et al. 2003, s.1)

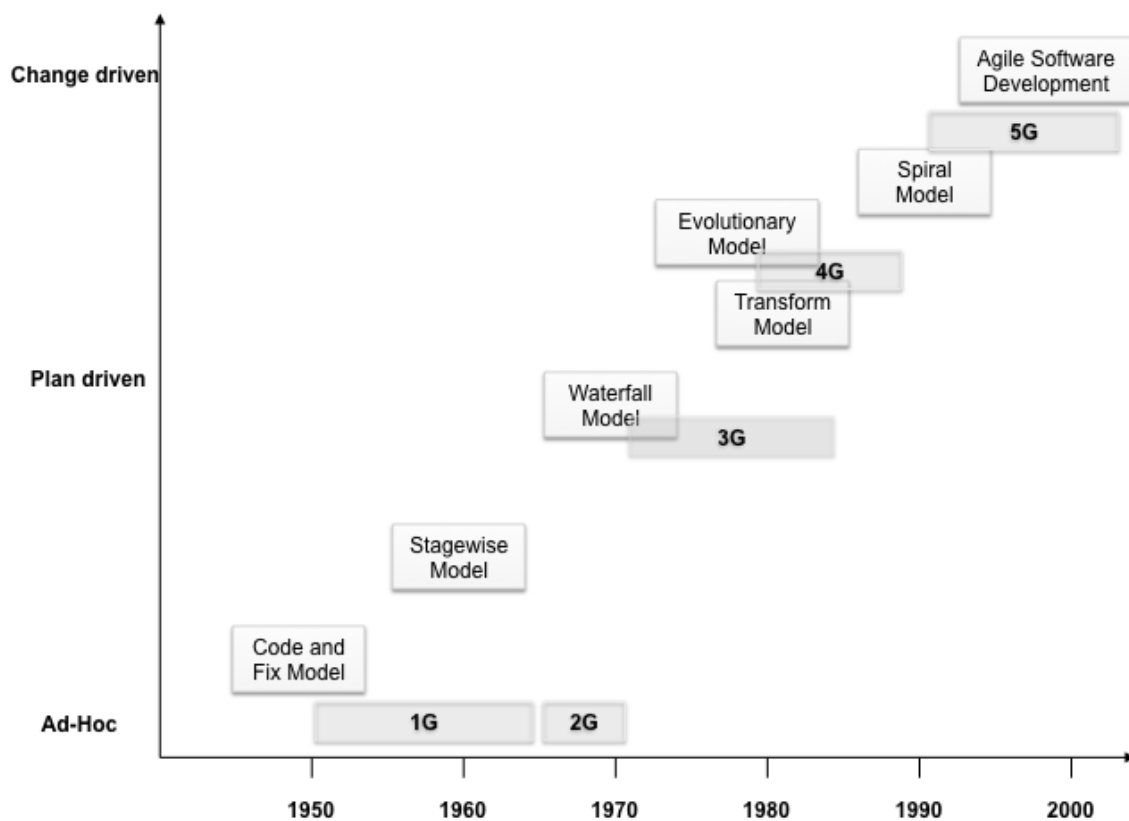
Lean periaatteiden mukaisesti kaikessa toiminnassa pitää pyrkiä välttämään hukkaa ja poistamaan se mikäli sitä havaitaan. Hukalla tarkoitetaan esimerkiksi erilaisten monimutkaisten hyväksymiskäytäntöjen purkamista tai päällekkäisten projektien tekemistä. Royce (1970, 328s.) kertookin artikkelissaan vuodelta 1970, että ohjelmistokehityksessä on kaksi aivan välttämätöntä vaihetta analysointi ja sitä välittömästi seuraava koodaus. Poppendieck korostaakin, että Lean periaatteiden mukaisesti voidaan tulkita, että vesiputousmallin kaikki muut vaiheet ovat turhia ja niistä voitaisiin hankkiutua eroon. (Poppendieck et.al, 2003, s.1-7)

Poppendieck et al. (2003, 4s.) määrittelee Lean -periaatteen mukaisen hukan ohjelmistokehitysprosessissa seuraavasti:

- partially done work (melkein valmis työ)
- extra processes (ylimääräiset prosessit, byrokratia)
- extra features (lisäominaisuudet)
- task switching (jatkuva työn vaihto, päällekkäiset projektit)
- waiting (odottaminen)
- motion (ihmisten tai tiedon turha liike) ja
- defects (virheet).

6 Innovaatioprosessin ja järjestelmäkehitysprosessin synteesi

Innovaatioprosessilla ja järjestelmäkehitysprosessilla näyttää olevan samankaltainen kehityskaari ajan ja ohjautuvuuden muodostamilla akseleilla. Kun Salon (2006) evoluutiomalliin lisätään innovaatioprosessin viisi sukupolvea, huomataan, että samankaltaisesti ohjautuvat mallit ovat kehittyneet hyvin samoihin aikoihin. Ketterän ohjelmistokehityksen ja 5. sukupolven innovaatioprosessin välillä näyttää kuitenkin olevan eroa ketterän ohjelmistokehityksen korkeamman muutosohjautuvuuden osalla. Innovaatioprosessin ja järjestelmäkehitysprosessin evoluutiot on esitetty kuvassa 14. Ohjelmistopuolen lähes valmistakin tuotetta on ehkä mahdollista muuttaa helpommin, kuin tuotekehitysprosessin loppupään ”kiinteää” tuotetta. Tämän perusteella oletamme Ketterien menetelmien olevan korkeammin muutosohjautuvia, kuin innovaatioprosessin 5G-malli. Alla olevaan kuvaajaan innovaatioprosessin kehitysvaiheet on sijoitettu kirjoittajien oman arvion perusteella. Ajanjaksot ovat lähdeteksteistä, mutta muutosohjautuvuus -akselin sijoitus eri sukupolville on tehty arvioimalla innovaatioprosessin eri sukupolvien ominaispiirteitä.



Kuva 14. Innovaatioprosessin ja järjestelmäkehitysprosessin evoluutio.

Kun tarkastellaan innovaatioprosessin ja vesiputousmallin ongelmia, huomataan seuraavat yhtymiskohdat:

- raskas dokumentaatio,
- prosessin aikaisten muutosten hallinta,
- tiedon siirtäminen tiimien ja jäsenten välillä
- ja pyrkimys tai paine kiinteisiin määrittelyihin projektien koon, aikataulun tai kustannusten suhteen (epävarmuuden hallinta).

Ongelmista voidaan huomata, että vesiputousmallin ongelmat ovat hyvin linjassa agile manifesto:n vastakkainasettelun kanssa. Voidaankin lähteä siitä oletuksesta, että löydetyt ongelmat edustavat merkittävää osajoukkoa vesiputousmallin kritiikistä, jota Agile Foundation manifestillaan pyrkii korjaamaan.

Taulukossa 2 on käsitelty yhtenevät ongelmakohdat tapauskohtaisesti ja esitelty ketterän ohjelmistokehityksen ratkaisu oman erikoisalansa ongelmiin.

Taulukko 2. Vesiputousmallin ja innovaatioprosessin ongelmien yhtenevyys.

Ongelma vesiputouksessa	Ongelma innovaatioprosessissa	Ratkaisu ketterissä menetelmissä	Työkalu
Raskas dokumentaatio	Raskas dokumentaatio	vaatimusten kuvaaminen tarinoina, low tech dokumentointi	Info radiator, white board, planning game
Muutosten käsittely	Muutosten käsittely vaikeutuu prosessin loppua kohti	ominaisuuksien priorisointi asiakkaan kanssa, iteraatiot	planning game, backlog
Osaamisen siirtäminen	Osaamisen siirtäminen	yhteinen työtila, pienet kehitystiimit, jatkuva kommunikaatio tehokkaalla modaaliteetilla	war room, scrum, parityöskentely, lyhyet prosessin mukaiset, jatkuva asiakkaan läsnäolo
Kiinteät määrittelyt	Pyrkimys kiinteisiin määrittelyihin alkupään ja tuoteprosessin hallitsemiseksi. Epävarmuus asiakkaan tarpeista.	iteratiivinen, muutosohjautuva prosessi, toteutus pienissä käyttövalmiissa inkrementeissä	
	FFE sekavuus		

6.1 Alkupää

Kuten taulukosta huomataan, innovaatioprosessin alkupäähän kohdistuvaa ongelmaa ei ole sellaisenaan kuvattu vesiputouksmallin ongelmaksi. Hannola selittääkin ongelman olevan yhtenäisistä osista huolimatta erilaisessa käytettävässä sanastossa sekä määritelmissä. Samassa Hannola kuitenkin huomauttaa, että joitain innovaatioprosessin alkupään toimintoja ei ole järjestelmäkehitysprosessissa lainkaan ja päinvastoin. Innovaatioprosessin uusien mahdollisuuksien tunnistaminen ja analysointi sekä uusien ideoiden synnyttäminen eivät ole sellaisenaan järjestelmäkehitysprosessin osia vaan niiden oletetaan tulevan prosessin ulkopuolelta asiakkaalta. Prosessien alkupään eroista huolimatta molemmilla aloilla on huomattu prosessin kokonaisuuden parantuvan keskittymällä alkupään parantamiseen. Hannolan mukaan innovaation alkupää onkin yksi kaikkein kriittisimmistä ja merkittävimmistä vaiheista ohjelmistotuotteen taloudellisessa kehityksessä. (Hannola 2009, s.9-11)

Alkupäässä onnistuminen riippuu monesta asiasta samaan aikaan. Ilman erittäin innovatiivista ideaa on innovaatioprosessia vaikea viedä onnistuneesti loppuun saakka. Ideointiin tarvitaan laajan kokemusperän omaavia asiantuntijoita, jotka ovat avoimia ja luovia. Alkupään onnistumisessa tarvitaan siis erityisen osaavia henkilöitä, toimivaa henkilökemialla, toimivaa organisaatorakennetta ja myös ripaus ”onnea”. Alkupäästä löytyy kuitenkin selkeitä ongelmia, jotka ovat yleisiä innovaatioprosessille organisaatiosta riippumatta. On mielenkiintoista huomata, että ongelmat ovat pääpiirteissään samoja kuin järjestelmäkehityksen ongelmat. Ketteristä menetelmistä siis voisi löytyä apua innovaatioprosessin ongelmiin, mutta aivan suoraan ketteriä menetelmiä ei voi ”leikkaa-liimaa” menetelmällä ottaa osaksi innovaatioprosessia, koska ohjelmistokehitys ja tuotekehitys eroavat määrittävästi siinä, että ohjelmistopuolella tuote ei ole fyysisesti olemassa. Suuri osa ketterien menetelmien parannuksista järjestelmäkehitysprosessiin on kuitenkin erilaisia tiedon ja taidon organisointimalleja – ihmisten toimintatapojen systematisointia. Näillä toimintamalleilla voisi nähdä olevan käyttöä myös ohjelmistokehityksen ulkopuolella.

6.2 Epävarmuuden ja muutosten hallinta

Epävarmuus asiakkaan tarpeista ja tuotevaatimuksista, sekä tuoteprosessin resurssoinnista ja aikataulutuksesta on myös ongelma innovaatioprosessin alkupään ja tuoteprosessin välillä. Varsinkin tuotteen kiinteät ominaisuudet pyritään lyömään lukkoon hyvin varhaisessa vaiheessa, vaikka edes asiakas ei välttämättä vielä tarkkaan osaa määrittää haluamiaan lopullisia ominaisuuksia. Tuoteprosessin edetessä, mahdollisuudet vaikuttaa lopulliseen tuotteen ominaisuuksiin tulevat hankalammiksi ja kallistuvat koko ajan. Alkupään ongelmista selvittää sitä

paremmin, mitä huolellisemmin alun määrittelyt tehdään, mutta sen ei tarvitse tarkoittaa sitä, että koko tuoteprosessiosuus täytyy aikatauluttaa ja resurssoida lopullisesti. Alkupää vaatii aikaa onnistuakseen, mutta samalla tuotteella on usein kiire markkinoille, joka johtaa ristiriitaan. Tuntuu oudolta, että vaikka asiakas ei osaisi tarkkaan määrittää lopullista tuotetta, pitäisi koko tuotekehitysprosessi eli valmistusaikataulu ja kiinteät ominaisuudet pystyä lyömään lukkoon.

Ketterissä menetelmissä kyseinen ongelma on ratkaistu niin, että kiinteää toteutuslaajuutta ei ole. Kiinteän toteutuslaajuuden puuttuminen ei tarkoita sitä, että prosessilla ei olisi loppua ollenkaan, vaan sitä että muutenkin epävarmaa tulevaisuutta ei pyritä ennustamaan ja hallitsemaan. Usein voi nimittäin käydä niin, että ajatuksen tasolla etukäteen määritellyn tuoteprosessin ajatellaan kulkevan ongelmitta, kun se kerran on määritelty kunnolla, mutta harvoin yksikään tuotekehitysprosessi viedään läpi ilman ongelmia. Asiakas ja kehitystiimi ovat jatkuvassa vuorovaikutussuhteessa, jolloin yhdessä priorisoidut vaatimukset toteutetaan inkrementaalisenä kehitysprosessina. Iteratiivinen kehitys tarjoaa mahdollisuuden tarkistaa prosessin tuotoksia lyhyin väliajoin ja ketterien menetelmien sisältämä asiakkaan läsnäolo tuo prosessiin mahdollisuuden tarkistaa asiakkaan tarpeet säännöllisin väliajoin. Mikäli asiakkaalle tulee uusia tarpeita kehityksen aikana, ne päätyvät luonnollisella tavalla arvioitavaksi seuraavaan iteraatioon valittavien tehtävien yhteydessä. Beckin (1999a) mukaan asiakas poimii viikon - kahden sisällä tapahtuvassa seuraavan iteraation suunnittelussa edelleen kiireellisenä pidetyt ominaisuudet toteutettaviksi. Ketterissä menetelmissä tuotekehitys on jatkuva muutosten mukana ohjautuva prosessi.

On tietenkin eri asia lähteä kehittämään nykyaikaista ohjelmistotuotetta, kuin kiinteää fyysistä tuotetta, johon on vaikea tehdä radikaaleja muutoksia kesken tuotekehitysprosessin – jopa pienien lisäyksien tekeminen tuotteeseen voi olla vaikeasti toteutettavissa. Nykyaikana kuitenkin suurin osa innovatiivisista tuotteista sisältää paljon teknologisia ratkaisuja, joista suuri osa pohjaa pelkästään ohjelmistokehitykseen, joten ainakin tällaisissa tapauksissa ketterien menetelmien ratkaisu sopii innovaatioprosessin ongelmaan.

6.3 Raskas dokumentaatio

Apilon ja Taskisen mukaan (Apilo et al. 2007. s. 160) useat (suomalaiset) yritykset ovat alkaneet kuvata tuoteprosessia prosessikuvauksilla, mutta samoin kuin ohjelmistokehitysprosessissa ongelmaksi on muodostunut määrittely- ja suunnitteluvaiheiden raskas dokumentaatio. Myös dokumenttien määrä ja kattavuuden puute on havaittu ongelmaksi.

Innovaatioprosessi ja järjestelmäkehitysprosessi yhtenevät selvästi raskaan dokumentaation osalta ja ketterät menetelmät on ratkaissut ongelman niin sanotuilla kevyillä käyttäjäkuvauksilla. Yhden ketterän menetelmän, XP:n, työkalu on asiakkaan vaatimusten muuttaminen käyttäjätarinoiksi siten, että asiakkaalle tuotettava arvo ja sisältö pystytään kuvaamaan sekä asiakkaan, että ohjelmoijan ymmärtämällä tavalla.

Asiakkaan pitäminen lähellä koko innovaatioprosessin läpi varmasti myös vähentää sitä tilannetta, että dokumentoinnista tulee itsetarkoitus. Dokumentoinnin tavoitteen tulisikin olla ensisijaisesti se, että saadaan selkiinnytettyä asiakkaan haluamat ominaisuudet niin, että ne ovat kaikkien ymmärrettävissä.

6.4 Tiedon siirtäminen

Yhtenä haasteena sekä innovaatioprosessissa että järjestelmäkehitysprosessissa nähdään tiedon ja osaamisen siirtäminen poikkifunktionaalisten tiimien välillä. Tiedon ja uuden osaamisen välittämisen tärkeyttä ei usein painoteta tarpeeksi. Yritysten sisällä tapahtuu myös usein päällekkäin tekemistä, väärinymmärryksiä ja väärin piirustus- ja kehitysversioiden aiheuttamaa ylimääräistä työtä, jotka johtuvat kommunikaatiokatkoksista.

Ketterissä menetelmissä ongelmaan on useitakin ratkaisuja. Mobile-D:ssä kommunikaation parantamiseksi on otettu käyttöön niin sanottu ”war room” eli huone jonne koko ohjelmistokehitys keskitetään. Keskitetyssä ohjelmistokehityksessä on myös kaikki tuotettu materiaali ja kehitykseen vaaditut dokumentit. Monissa menetelmissä käytetään prosessin etenemisen ja tilan kuvaamiseen ”Information Radiator” – taulua, johon yksinkertaisilla lapuilla voidaan merkitä työn alle otetut ja keskeneräiset tehtävät sekä valmistuneet ominaisuudet ja työvaiheet. Taulussa näkyy myös tehtävät, jotka on tunnistettu, mutta joita ei ole vielä otettu toteutettavaksi. Tällöin kaikki oleellinen tieto on yhdessä huoneessa.

Menetelmää voisi soveltaa innovaatioprosessin ainakin suurimmilta osin – ainakin tuotekehityksen kaikki dokumentaatio ja materiaali voisi olla keskitettynä yhteen huoneeseen. Myös ”Information Radiator” –taulun tyyppistä ratkaisua olisi mahdollista käyttää innovaatioprosessissa. Suurin ongelma on ehkä henkilöstön sijoittaminen yhteiseen tilaan tuotekehitysprosessin ajaksi, mutta ainakin tietokoneella tehtävät työvaiheet, innovaation alkupää ja asiakastapaamiset olisi mahdollista ainakin osittain toteuttaa yhteisessä tilassa. Yhteisen tilan käyttäminen olisi mahdollista toteuttaa

ainakin niin, että olisi huone, jossa tuotekehitysprojektin sen hetkinen tilanne ja materiaali olisivat helposti saatavilla.

Tiedon välitystä lisää myös XP:ssä käytettävä parityöskentely ja Scrumissa käytettävä päivittäinen ja viikoittainen kokoontuminen. Parityöskentely mahdollistaa hiljaisen tiedon siirtymisen ja toistuvat lyhyet kokoontumiset auttavat osallistujia keskittymään tekeillä oleviin tehtäviin ja ratkaistaviin ongelmiin. Molemmilla työtavoilla on mielestämme päällekkäistä työtä ehkäisevä ja myös ongelmien piilottelua ehkäisevä vaikutus. Kun ongelmien ratkaisemiseen osallistuu suurempi joukko, se tehostaa suorituskykyä ja korottaa yhteishenkeä.

Varsinkin nykyteknologian aikana tuntuu oudolta, että päällekkäin tekeminen ja kommunikaatiokatkokset on nostettu yhdeksi isoksi ongelmaksi innovaatiota tutkivassa kirjallisuudessa (Apilo et al. 2007, s. 163). Nykyaikaisilla intranet –järjestelmillä ja loputtomilla tiedonsiirtomahdollisuuksilla luulisi olevan mahdollista luoda vaikka ketteristä menetelmistä tuttu ”war room” virtuaalisena tilana, jonne innovaatioprosessin osalliset voisivat päivittää oman työnsä tilaa. Ongelmaa selittänee osaltaan Ståhlen (Ståhle & Grönroos, 2000, s.106) käsitys ettei hiljaista tietoa kyetä välittämään tietoteknisin välinein, vaan se välittyy parhaiten suullisessa kommunikaatiossa ja työskentelemällä yhdessä.

7 Johtopäätökset

Boehm et.al (2005) odottaa ketterien menetelmien siirtyvän muihinkin organisaatioihin, missä on kehittynyt ja kypsä suunnitelmaohjautuva prosessi. VTT:n julkaisema Mobile-D menetelmä yhdistelee eri ketterien menetelmien ominaisuuksia uudeksi kokonaisuudeksi ja samalla osoittaa, että ominaisuudet ovat menetelmästä irrallisia ja siirrettäviä siten, että niiden hyödyt säilyvät.

Boehmin et. al (2005) mukaan ketterä ohjelmistokehitys vahvistaa ja päivittää prosessissa hiljaista tietoa (tacit knowledge) enemmän kuin kirjoitettua tietoa (explicit knowledge). Stählen et al.:n mukaan organisaation kaikkein arvokkain tieto on säilyttänyt sen työntekijöiden hiljaisen tiedon varastoihin. Hiljaista tietoa on mahdoton siirtää muuten kuin kahdenkeskisessä kanssakäymisessä mutta sen siirtymistä voidaan tehostaa kuvien, kaavioiden tai mallien käyttämisellä. (Stähle et. al, s.106-107)

Stählen et.al mukaan kaaos on välttämätön edellytys uuden tiedon, innovaatioiden tai ideoitten synnylle. Vaikka emme pysty hallitsemaan innovaatioiden syntyä, voimme luoda ympäristön, missä innovaatiot ovat todennäköisempiä kuin muulloin. Kun ymmärrämme, että kaaos on innovaatioiden tarvitsema kasvupohja, kysymys onkin enää vain siitä, kuinka se saadaan aikaan. (Stähle et.al, 2000, s.120-122)

Ketterien menetelmien työtapojen omaksuminen innovaatioprosessiin voikin vahvistaa sen suorituskykyä lisäämällä hiljaisen tiedon siirtymistä organisaation sisällä. Ketterien menetelmien työtapojen omaksumisen, kuten yhdessä tilassa työskentelyn ja tehokkaan suullisen kommunikaation, tuloksena voidaankin vahvistaa organisaation kehittymistä Stählen kolmiportaisella asteikolla. Mekaanisesta, kirjoitettua tietoa hyödyntävästä organisaatiosta, orgaanisen, hiljaista tietoa hyödyntävän kasvuvaiheen kautta kohti dynaamista intuitiiviseen eli potentiaaliseen tietoon pohjautuvaa organisaatiota. Samalla organisaation kompetenssi ja kilpailuasema markkinoilla kasvaa. (Stähle et.al, 2000, s.93-127)

Innovaatioprosessin ongelmiin löytyy ketteristä menetelmistä mielestämme ainakin muutamia konkreettisia ratkaisuja;

- Asiakastarpeen ymmärtämien ja muovaaminen tuotevaatimuksiksi on koettu vaikeaksi ja samalla ristiriitaiseksi tavoitteeksi tuoteprosessin nopean aloittamisen kanssa. Ongelmia on

havaittu myös tuoteprosessin aikataulutuksessa ja resurssoinnissa. Muutosohjautuvien ohjelmistokehitysmenetelmien ratkaisu on ollut luopua kiinteästä toteutuslaajuudesta. Korvaamalla laaja ja usein käyttämättä jäävä etukäteissuunnittelu jatkuvalla vuorovaikutussuhteella asiakkaan ja kehitystiimin välillä on mahdollistanut inkrementaalisen tuoteprosessin. Inkrementaalinen prosessi tuottaa itsearviointia tasaisin väliajoin, jolloin voitaisiin peilata tuoteprosessin tilaa myös emergentteihin asiakkaan tärkeinä pitämiin ominaisuuksiin ja yhteisellä päätöksellä lisätä ne mukaan tuoteprosessin seuraavaan inkrementtiin.

- Ratkaisuna innovaatioprosessin ongelmaan raskaan dokumentaation osalta voisi olla ketterien menetelmien kevyiden käyttäjäkuvausten eli käyttäjätarinoiden soveltaminen. Asiakkaan vaatimukset muutetaan käyttäjätarinoiksi siten, että asiakas ja kehitystiimi ymmärtävät ongelman samalla tavalla. Asiakkaan pitäminen lähellä mielestämme myös vähentäisi sitä vaaraa, että dokumentoinnista tulee liian pikkutarkkaa vailla ilmeistä lisäarvoa asiakkaalle tai yritykselle.
- Tiedon ja osaamisen siirtäminen nähdään yhtenä suurimpana haasteena innovaatioprosessissa. Kommunikaatiokatkokset ja väärinymmärrykset aiheuttavat ylimääräistä työtä. Ketterissä menetelmissä hyödynnetään useita työskentelytapoja, jotka parantavat tiedon ja osaamisen siirtymistä. War room eli yhteinen tila, josta löytyy projektin henkilökunta, työstettävä materiaali ja samassa tilassa tai sen välittömässä läheisyydessä myös Information Radiator, josta selviää meneillään olevan vaiheen tila nopealla vilkaisulla. Tilan ei välttämättä tarvitsisi olla fyysinen vaan se voisi olla myös yrityksen tai yhteistyökumppanien muodostaman verkon yhteisessä tietojärjestelmässä. Myös parityöskentely, sekä päivittäinen stand-up meeting ja viikoittaiset kokoontumiset voisivat parantaa tiedon ja osaamisen siirtämistä.

Ainakin osassa nykyaikaisista innovaatioprosesseista käytetään niin sanottua porttimallia (stage gate) tuoteprosessin ohjaamiseen. Porttimallin tarkoituksena on arvioida prosessin kulkua tasaisin väliajoin, jotta pystytään ehkäisemään virheiden kumuloituvaa vaikutusta loppuprosessissa. Tuoteprosessiin sovitaan pysähdyspaikkoja, jossa pitäisi tarkastella projektin tilaa ja päättää jatkamisesta, takaisin palaamisesta tai projektin lopettamisesta. On kuitenkin todettu, että liian usein

yritykset suhtautuvat portteihin vain muodollisina vaiheina, jotka käydään läpi, jotta tuoteprosessia voidaan jatkaa ”junan” tavoin. (Apilo et al. 2007, s. 164) Todellista arviointia ei siis tapahdu.

Ajatukseksi nousi, että porttimallia voisi soveltaa siten, että se onkin asiakkaan tekemää ominaisuuksien priorisointia ja toteutuneiden ominaisuuksien arviointia. Tällöin päästäisiin irti ongelmasta, missä tuotekehitystiimi on haluton arvioimaan omaa tekemistään tarpeeksi kriittisesti. Koska toteutettavat tuotevaatimukset on priorisoitu yhdessä asiakkaan kanssa, portilla tapahtuva arviointi ei kohdistuisi vain kehitystiimiin. Asiakas toimisi tässä mallissa portinvartijana, jolloin oletamme, että asiakkaan haluamat ominaisuudet todella toteutuisivat tärkeysjärjestyksessä. Asiakas varmasti myös osaisi keskeyttää projektin, jos huomaisi sen olevan kannattamaton esimerkiksi markkinoiden muutoksen tai omien vaatimustensa johdosta.

Tutkiessamme innovaatioprosessin ja järjestelmäkehitysprosessin kehityskulkuja huomasimme, että yhteneviä ongelmia löytyi selkeästi neljä kappaletta; raskas dokumentaatio, muutosten ja epävarmuuden hallinta, sekä poikkifunktionaalisen tiedon siirto. Ketterien menetelmien soveltaminen jokaiseen näistä ongelmista vaatii jatkotutkimista, jotta saadaan kokonaisvaltainen kuva siitä, kuinka täydellisesti ketteriä menetelmiä voi soveltaa innovaatioprosessiin. Työn laajuus estää syvemmän ketterien menetelmien soveltamisen tutkimisen, mutta johtopäätöksenä voidaan todeta, että ketteristä menetelmistä varmasti löytyisi osia, joita voi soveltaa innovaatioprosessiin.

Työn tarkoitus oli selvittää löytyykö prosesseista yhteneviä piirteitä ja tutkia alustavasti, voisiko ketteristä menetelmistä löytyä niihin ratkaisuja. Työn laajuudesta johtuen syvempi ratkaisujen ja ketterien menetelmien soveltamisen tutkiminen ei kuitenkaan ole mahdollista. Työ herättää selkeitä jatkotutkimuksen kohteita: Kuinka kokonaisvaltaisesti ketterät menetelmät ovat siirrettävissä innovaatioprosessiin? Tuovatko ketterät menetelmät selviä kustannussäästöjä? Löytyisikö ketteristä menetelmistä apua erityisesti palveluinnovaatioprosessin parantamisessa?

8 Yhteenveto

Perinteisen ohjelmistokehitysprosessin eli vesiputousmallin ja innovaatioprosessin vaiheet ovat hyvin samankaltaisia. Vesiputousmallin soveltamista käytäntöön on vaikeuttanut sen kaavamaisuus ja joustamattomuus. Ongelmien korjaamiseksi vesiputousmallia on kehitetty eteenpäin ja viimeisimpänä kehitysaskeleena on muodostunut joukko menetelmiä, joita kutsutaan ketteriksi menetelmiksi. Ketteriä menetelmiä yhdistää pyrkimys nopeaan asiakasarvoon ja käyttökelpoisen ohjelmistoversion tuottamiseen inkrementaalisen ohjelmistokehitysmallin avulla.

Työn tarkoitus oli lähteä tutkimaan voidaanko innovaatioprosessissa hyödyntää tietojärjestelmien kehittämisessä havaittuja ketteriä menetelmiä joko kokonaan tai osittain. Lähdimme analysoimaan järjestelmäkehitysprosessin ja innovaatioprosessin keskinäisiä yhtäläisyyksiä, eroja ja ongelmia. Tavoitteena oli prosessien yhtäläisyyksien ja ongelmien määrittämisen jälkeen tutkia, löytyisikö innovaatioprosessin ongelmiin apua ketteristä ohjelmistokehitysprosesseista tai yksittäisistä ketteristä menetelmistä.

Vesiputousmalli kehittyi 70-luvulla ensimmäisen ohjelmointikriisin herättämänä ja sen kehittymisen ajurina toimi Yhdysvaltain johtaman puolustusliitto NATOn koolle kutsumat konferenssit. Näiden tuloksena varhaista tilamallia parannettiin tunnistamalla tarve vuorovaikutukseen asiakkaan kanssa prosessin kuluessa ja pyrkimällä parantamaan prosessin eri vaiheiden välistä vuorovaikutusta. Vesiputousmalli ei kuitenkaan ratkaissut ohjelmistotuotannon ongelmia ja sitä seurasi joukko muita menetelmiä, joiden avulla pyrittiin korjaamaan sen pahimpia puutteita. Pyrkimys kiinteisiin määrittelyihin, sekä kustannus- ja työmääräarvioihin sekä kyvyttömyys hallita muuttuvia vaatimuksia ja menetelmän vaatimaa raskasta dokumentointia tunnistettiin vesiputousmallin pahimmiksi ongelmiksi 90-luvulle tultaessa.

Innovaatioprosessi on kehittynyt 1950- 1970 luvun lineaarisista markkinavetoisista ja teknologiatyöntöisistä malleista monimutkaisiin toiminnot integroiviin verkostomalleihin, jotka ovat muutosohjautuvia ja innovaation katsotaan olevan jatkuvaa läpi prosessin. Innovaatioprosessin kehityksessä on nähtävissä viisi sukupolvea, joilla on toisistaan selkeästi eroavat ominaispiirteet. Varhaisten lineaaristen mallien ongelmat ovat kehittyneemmissä malleissa korjaantuneet, mutta tilalle on noussut uusia ongelmia, joista merkittävimmät liittyvät innovaatioprosessin alkupäähän ja konseptointivaiheeseen.

Nykyaikaisen innovaatioprosessin merkittävimmit ongelmat nousivat;

- raskas dokumentaatio,
- prosessin aikaisten muutosten hallinta,
- tiedon siirtäminen tiimien ja jäsenten välillä
- ja pyrkimys tai paine kiinteisiin määrittelyihin projektien koon, aikataulun tai kustannusten suhteen (epävarmuuden hallinta).

Yllä kuvatut ongelmat ovat yhteneviä järjestelmäkehitysprosessin ongelmien kanssa, joihin ketterät ohjelmistokehitysmenetelmät ovat tuoneet ratkaisun. Huomasimme myös, että innovaatioprosessi ja järjestelmäkehitysprosessi ovat kehittyneet lähes rinnakkain suhteessa aikaan, ad-hoc malleista kohti kehittyneitä muutosohjautuvia malleja.

Ketterät menetelmät tuovat yllä oleviin ongelmiin ratkaisuja, joten voidaan olettaa, että ketteristä menetelmistä olisi hyötyä innovaatioprosessin parantamisessa. Täysin ehdoitta ketteriä menetelmiä ei kuitenkaan innovaatioprosessiin voi soveltaa, koska ohjelmistokehitys ja tuotekehitys eroavat suuresti siltä osin, että tuotekehityksen lopputulos on kiinteä tuote, kun taas ohjelmistokehityksen lopputulos on koodia.

Ketterissä menetelmissä on kuitenkin paljon organisaation toimintatapoihin, tiedon ja osaamisen siirtoon, sekä asiakkaan tarpeiden ymmärtämiseen tuomia parannuksia, joita voitaisiin soveltaa nykyaikaiseen innovaatioprosessiin.

Lähteet

Abrahamsson, P., Salo, O. Ronkainen, J. & Warsta, J. 2002. Agile software development methods: Review and analysis, VTT publications. 478. 107s.

Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P. & Salo, O. 2004. MobileD: An Agile Approach for Mobile Application Development. In: The proceedings of the 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA '04). October 24-28, 2004. Vancouver, British Columbia, Canada. s.174-175.

Apilo, T., Taskinen, T. & Salkari, I. 2007. Johda Innovaatiota. Talentum Media Oy. 260 s.

Apilo, T. & Taskinen, T. 2006. Innovaatioiden johtaminen. VTT, Espoo. 112 s. [saatavissa: <http://www.vtt.fi/inf/pdf/tiedotteet/2006/T2330.pdf>]

Beck, K. 1999. Embracing Change With Extreme Programming. IEEE Computer. Vol. 10(33), October, s.70-77.

Boehm, B. 1988. A Spiral Model of Software Development and Enhancement. Computer, Vol. 21, 5. Toukokuu 1988. s.61-72.

Boehm, B. & Turner, R. 2005. Management Challenges to Implementing Agile Processes in Traditional Development Organizations. IEEE Software. Vol. 22(5), September-October, s.30-39.

DeGrace P. & Stahl L., 1990, Wicked Problems, Righteous Solutions, Prentice-Hall. Upper Saddle River, NJ. 176s.

Cockburn, A. 2003, Agile Software Development, Pearson Education Inc, Arlington Street. 222s.

Galanakis, K. 2006. Innovation Process. Make sense using systems thinking. Technovation. Vol 26, nro 11, s. 1222-1232.

- Kahn, Kenneth B. (toim.). 2005. PDMA Handbook of New Product Development (2. ed.). John Wiley & Sons. 618 s.
- Larman, C. & Basili, V. 2003 Iterative and Incremental Development: A Brief History. IEEE Computer Society. June, s.47-56.
- NATO Science Committee. 1969. Software Engineering. Report of the Conference Sponsored by the NATO Science Committee. Naur, P. & Randell B. Scientific Affairs Division, NATO. Garmish 7-11 Lokakuu, 1968. 136s.
- Poppendieck, M. & Poppendieck, T. 2003. Lean Software Development: An Agile Toolkit. Addison-Wesley. Upper Saddle River. 203s.
- Rothwell, R. & Gardnier, P. 1985. Invention, innovation, re-innovation and the role of the user: A case study of British hovercraft development. Technovation. Vol 3, nro 3, s. 168.
- Rothwell, R. 1994. Towards the fifth-generation innovation process. International Marketing Review 1994. Vol 11, nro 1, s. 7-31.
- Royce, W. W. 1970. Managing the Development of Large Software Systems. In: The proceedings of the WESCON. San Francisco. IEEE CS. s. 328-339.
- Salo, O. 2006. Enabling Software Process Improvements in Agile Software development Teams and Organisations, VTT publications. 618. 149s.
- Schwaber, K. 1995. Scrum Development Process. OOPSLA'95 Workshop on Business Object Design and Implementation. Springer-Verlag. 23s.
- Ståhle, P. & Grönroos, M. 2000. Dynamic Intellectual Capital, Knowledge Management in Theory and Practice. WSOY, Helsinki. 296s.
- Takeuchi H. & Nonaka I., 1986, New Product Development Game, Harvard Business Review. January-February, s. 2-7.

Tidd, J. Bessant, J. & Pavitt, K. 2005. Managing Innovation 3rd Edition: Integrating technological, market and organizational change. John Wiley & Sons, Ltd. 582 s.

Womack, J., Jones D. & Roos, D. 1991. The Machine that Changed the World: The Story of Lean Production. HarperPerennial. 328s.

Internet –lähteet:

Innovation Unit, UK Department of Trade and Industry. 2004. What is Innovation? [WWW-sivu]. [viitattu 4.3.2010]. Saatavissa: <http://www.innovation-unit.co.uk/about-us/what-is-innovation/what-is-innovation.html>

Tilastokeskus. EU:n Innovaatiotutkimus 2000. [WWW-sivu]. [viitattu 11.04.2010]. Saatavissa: http://www.stat.fi/til/inn/2000/inn_2000_2004-05-31_kat_001.html

Abrahamsson, P. 2006. Efficient Development of Mobile Applications: Test Early, Test Often and Keep Testing. [PDF-dokumentti]. [viitattu 13.4.2010]. Saatavissa: http://www.ouka.fi/wirelesscities/Presentations/abrahamsson_wireless_public.pdf

Komulainen, K. Mäntyniemi, A. Pikkarainen, M. 2006. Agile Software Development: Summary of Kotel Workshop. [PDF-dokumentti]. [viitattu 13.4.2010]. Saatavissa: http://www.agile-itea.org/public/deliverables/ITEA-AGILE-D2.8_v1.0.pdf

Liite 1. Kuvat

Kuva 1. Innovaatioavaruus. (Tidd et al. 2005, s. 13) s.4

Kuva 2. Nykyaikaisen innovaatioprosessin jako alkupäähän, tuotekehitykseen ja lanseeraukseen. (Kahn 2005, s. 82) s.5

Kuva 3. Innovaatioprosessin yksinkertainen perusmalli. (Tidd et al. 2005, s. 67) s.7

Kuva 4. Innovaatioprosessin evoluutio Rothwellin mukaan. (Rothwell 1994, s. 7) s.9

Kuva 5. Teknologia-työntöinen innovaatioprosessi. (Rothwell 1994, s. 7) s.10

Kuva 6. Markkinavetoinen innovaatioprosessi. (Rothwell 1994, s. 8) s.11

Kuva 7. Kolmas eli interaktiivinen sukupolvi. (Rothwell 1994, s. 10) s.13

Kuva 8. Neljäs eli toiminnot integroiva sukupolvi. (Rothwell 1994, s.12) s.14

Kuva 9. 5G:n tärkeimmät strategiset pääkohdat. (Galanakis 2006, s. 1224) s.15

Kuva 10. Innovaatioprosessi sisältäen alkupään, konseptoinnin ja tuotekehitysprosessin. (Apilo et al. 2006, s. 43) s.17

Kuva 11. Vesiputousmalli (Boehm, 1988, s.62) s.21

Kuva 12. Ohjelmistokehitysmallien evoluutio (Salo, 2006, s.18) s.22

Kuva 13. Information Radiator (Abrahamsson, 2006, s.19) s.27

Kuva 14. Innovaatioprosessin ja järjestelmäkehitysprosessin evoluutio. s.29