



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Scrumin käyttö sovelluskehitysprojektissa - Case: Oy Samlink Ab:n sähköisen työpöydän pi- lottiprojekti

Roukala, Selja

2011 Leppävaara

Laurea-ammattikorkeakoulu
Leppävaara

Scrumin käyttö sovelluskehitysprojektissa - Case: Oy Samlink Ab:n sähköisen työpöydän pilottiprojekti

Selja Roukala
Tietojärjestelmäosaamisen koulu-
tususohjelma
Opinnäytetyö
Kesäkuu, 2011

Sisällys

1	Johdanto.....	7
1.1	Oy Samlink Ab.....	7
1.1.1	Sähköisen työpöydän pilottiprojekti	8
1.2	Työn tavoitteet ja tärkeys.....	8
1.3	Tutkimusongelma ja -kysymykset	8
1.4	Rajaukset	9
1.5	Käsitteet ja niiden selitykset	9
2	Teoreettinen tausta	10
2.1	Perinteinen sovelluskehitys ja perinteiset menetelmät.....	10
2.1.1	Vesiputousmalli	10
2.1.2	Spiraalimalli	11
2.1.3	Rational Unified Process	11
2.2	Ketterä sovelluskehitys	12
2.3	Ketterät sovelluskehitys -menetelmät	13
2.3.1	Extreme programming (XP).....	14
2.3.2	Adaptive software development (ASD)	14
2.3.3	Agile modeling (AM).....	15
2.4	Perinteiset menetelmät vastaan ketterät menetelmät	15
2.5	Ketterä Scrum -menetelmä.....	17
2.6	Ketterän Scrum-menetelmän käytännöt ja niiden käyttö sovelluskehityksessä.....	20
2.6.1	Sprint Planning Meeting	20
2.6.2	Product Backlog	21
2.6.3	Sprintti	22
2.6.4	Daily Scrum Meeting	24
2.6.5	Sprint Review	25
2.7	Ketterän Scrum - menetelmän roolit	26
2.7.1	Product Owner	26
2.7.2	Scrum Master	27
2.7.3	Scrum-tiimi	29
3	Tutkimusmenetelmän kuvaus	31
3.1	Tapaustutkimus (case study)	31
3.2	Tutkimuksen toteutus	32
3.2.1	Tutkimussuunnitelma	32
3.2.2	Tapauksien ja toimintaympäristön kuvaus.....	33
3.2.3	Tutkimuksen suunnittelu.....	34
3.2.4	Tutkimuksen valmistelu	35
3.2.5	Aineiston kerääminen	36

	3.2.6 Aineiston analysointi.....	37
	3.2.7 Tuloksien jakaminen.....	40
4	Tulokset.....	41
	4.1 Yrityksen ja asiakkaan näkemykset	41
	4.2 Scrumin suurimmat edut ja hyödyt	43
	4.3 Scrum vs. perinteinen menetelmä	44
	4.4 Sprint Planning - tapaaminen	45
	4.5 Product Backlog	45
	4.6 Sprintit	45
	4.6.1 Sprint Backlog.....	46
	4.7 Product Owner	46
	4.8 Daily Scrum ja Scrum Master	47
	4.9 Sprint Review -tapaaminen.....	47
	4.9.1 Retrospective	48
	4.10 Organisaatio.....	49
	4.11 Muut huomioon otavat asiat	50
	4.12 Kehittämisiä ja havaintoja	50
5	Yhteenveto	51
	5.1 Vastaaminen tutkimusongelmaan ja tutkimuskysymyksiin.....	51
	5.1.1 Vastaaminen tutkimusongelmaan	52
	5.1.2 Vastaaminen tutkimuskysymyksiin	52
	5.2 Yhteenveto tutkimustuloksista	54
	5.3 Keskustelua tutkimuksesta	55
	5.4 Ehdotuksia jatkotutkimukseksi	56
	Kuvat	60
	Taulukot	61
	Liitteet.....	62

Selja Roukala

Scrumin käyttö sovelluskehitysprojektissa - Case: Oy Samlink Ab:n sähköisen työpöydän pilottiprojekti

Vuosi 2011

Sivumäärä 63

Tämän opinnäytetyön päätavoitteena oli tutkia miten ketterän Scrum - menetelmän käyttöönotto ja käyttö onnistui Oy Samlink AB:n sähköisen työpöydän pilottiprojektissa ja mitä pitäisi jatkossa kehittää ja parantaa, jotta Scrumin käyttö olisi tulevissa Samlinkin sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa. Edellisiä selvitettiin projektissa mukana olleiden asiakkaiden ja muiden asiantuntijoiden avulla. Samalla haluttiin myös selvittää, mitä yritys ja asiakkaat odottavat ketterän-Scrumin käytöltä ja kohtasivatko yrityksen ja projektissa mukana olleiden asiakkaiden ja muiden asiantuntijoiden näkemykset toisensa sekä pystyttiinkö näitä näkemyksiä tukemaan teoreettisten näkemyksien avulla.

Tutkimusaineistona käytettiin empiirisen aineiston osalta Samlinkin Sovelluspalvelut yksikönjohtajan ja projektissa mukana olleiden asiakkaiden ja muiden asiantuntijoiden näkemyksiä sekä teoreettisen aineiston osalta aiheeseen liittyviä teoreettisia näkemyksiä. Empiirisen aineiston osalta näkemyksistä tehtiin yhteenveto ja niitä verrattiin toisiinsa niissä kohdin missä se oli mahdollista. Teoreettisen aineiston osalta yritettiin löytää ne väittämät, joiden avulla pystyttiin tukemaan empiirisiä näkemyksiä.

Tutkimuksen tulokset perustuvat edellä mainittuihin näkemyksiin. Yrityksen, asiakkaiden ja muiden asiantuntijoiden näkemyksistä saatuja tuloksia pystyttiin hyvin tukemaan teoreettisin väittämin. Saadut tulokset mahdollistivat aineistojen analyysien tekemisen ja yhdistettyjen tuloksien avulla saatiin vastaukset tutkimuskysymyksiin ja tutkimusongelmaan.

Tutkimusmenetelmäksi valittiin tapaustutkimus, koska se soveltuu tutkimusmenetelmäksi silloin kun, käytetään ja yhdistellään monia eri tietolähteitä ja halutaan löytää kehittämis- ja parannusehdotuksia johonkin asiaan.

Asiasanat: Scrum, ketterät menetelmät, sovelluskehitys, tapaustutkimus

Selja Roukala

The usage of Scrum in software development project - Case: Samlink Ltd.'s Electronic Desktop pilot project

Year	2011	Pages	63
------	------	-------	----

The main objective of this thesis was to investigate how the introduction and implementation of the agile process model Scrum succeeded in Samlink Ltd.'s Electronic Desktop's pilot project, and what needs to be improved and develop, so that the usage of Scrum would be easy and flexible in future projects. Investigation was done together with customers and other experts who took part in the project. At the same time there was a need to find out what the company and customers expected from the agile Scrum process, and did the expectations meet between the company, customers and other experts that took part in the project, and finally, did the views of the project participants and the theory of the Scrum process meet.

Both empiric and theoretical research material was used. The views of the director of Samlink's Software Services unit, customers and other experts constituted the empiric research material. Views were summarized, and the findings were compared to each other whenever possible. Theoretical research material was used to back up the empiric findings.

The results of this study are based on the views of company, customers and other experts. Theoretical research material supported the findings of company, customers and experts very well. Findings enabled the analysis of the research material, and the combined results were used to get answers to the research questions and problems.

Case study was chosen as research method, since it suites as research method when many information sources are combined and used as a source for improvement and development.

Keywords: Scrum, agile methods, software development, case study

1 Johdanto

Opinnäytetyön tarkoituksena on selvittää miten ketterän Scrum -menetelmän käyttöönotto onnistui Samlinkin sähköisen työpöydän pilottiprojektissa ja mitä pitäisi jatkossa ottaa huomioon jotta ketterän Scrum -menetelmän käyttö olisi tulevaisuudessa Samlinkin sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa. Samalla halutaan selvittää miksi ketterää Scrum -menetelmää päätettiin kokeilla Samlinkissa ja mitä yritys ja asiakkaat siltä odottavat. Erilaisina näkökulmina tarkastellaan kirjallisuuden, yrityksen sekä projektissa mukana olleiden asiakkaiden ja muiden asiantuntijoiden näkemyksiä. Opinnäytetyö sisältää luvut teoreettisesta taustasta, tutkimusmenetelmästä, tuloksista ja yhteenvedosta.

Tutkimusmenetelmänä käytetään tapaustutkimusta eli case studya. Tapaustutkimus valittiin menetelmäksi koska valittua kohdetta tutkitaan monesta eri näkökulmasta ja tietoa kerätään useiden lähteiden avulla (Yin 2009)

1.1 Oy Samlink Ab

Opinnäytetyö toteutetaan Oy Samlink Ab:lle. Samlink on finanssialan palveluihin erikoistunut yhtiö joka perustettiin vuonna 1994. Nykyään Samlink-konserniin kuuluu emoyhtiö Oy Samlink Ab ja tytäryhtiöt PP-Laskenta sekä Samcom Oy (Samlinkin vuosikertomus 2010, 3). Tämä tutkimus koskee Samlinkia.

Samlink-konsernin palveluihin kuuluu mm. pankki- ja rahoitusalan yritysten tietotekniikka-palvelujen kehittäminen ja ylläpito, monipuoliset taloushallinto- ja raportointipalvelut sekä sähköisten asiointipalvelujen ja portaalien järjestelmätoimittaminen (Samlinkin kotisivut 2011).

Samlink-konsernin asiakkaita ovat muun muassa Aktia, Säästöpankit, Paikallisosuuspankit, Handelsbanken Suomi, Suomen Hypoteekkiyhdistys, Itella IPS, Elisa ja Helsingin Energia (Samlinkin vuosikertomus 2010, 3.)

1.1.1 Sähköisen työpöydän pilottiprojekti

Viime vuoden marraskuussa käynnistyi sähköisen työpöydän pilottiprojekti, missä rakennetaan pankkien toimihenkilöiden käyttöön sähköistä työpöytä. Sähköinen työpöytä kuuluu Samlinkin NEXT-hankeeseen. Hankkeen tarkoituksena on mahdollistaa kaikkien pankkien operatiivisten järjestelmien käyttäminen yhden käyttöliittymän kautta. (Samlinkin vuosikertomus 2010, 9 - 10).

1.2 Työn tavoitteet ja tärkeys

Opinnäytetyön ensisijaisena tavoitteena on selvittää miten ketterän Scrum - menetelmän käyttöönotto ja käyttö onnistui Samlinkin sähköisen työpöydän pilottiprojektissa ja mitä jatkossa pitäisi ottaa huomioon, jotta tulevissa Samlinkin sovelluskehitysprojekteissa Scrumin käyttöönotto ja sen käyttö olisi mahdollisimman helppoa ja joustavaa. Scrumin käyttöönotto ja käytön onnistumista tutkitaan asiakkaiden ja projektissa mukana olleiden muiden asiantuntijoiden avulla.

Toisena tavoitteena on kasvattaa omaa tietämystä ketterän Scrum - menetelmän ideologiasta ja näin ollen selvittää sen käytännöt, sen käyttö sovelluskehityksessä sekä sen eroavaisuus perinteisiin menetelmiin.

Opinnäytetyö on tärkeä, koska sen avulla Samlink yrityksenä ja sen henkilöstö saavat näkemystä siihen miten ketterän Scrum - menetelmän käyttö koettiin sähköisen työpöydän pilottiprojektissa ja mitä täytyy ottaa huomioon tulevissa sovelluskehitysprojekteissa. Samlink yrityksenä saa myös tietoa siitä, mitä asiakkaat odottavat ketterältä-Scrumilta ja mitä hyötyjä ja etuja Samlink voi Scrumilta odottaa.

Lisäksi hyötynä tuotetaan yleistä tietoa siitä, mitä etuja ketterä Scrum -menetelmä tuo sovelluskehitykseen, mitä sen käyttöönotossa tulee huomioida ja miten se käyttöönotto tulisi tehdä, jotta se onnistuu parhaalla mahdollisella tavalla.

1.3 Tutkimusongelma ja -kysymykset

Samlinkin asiakkaat halusivat kokeilla ketterän Scrum -menetelmän käyttöä koska he kokivat aikaisemman systeemyömenetelmän kankeaksi ja joustamattomaksi. Ketterää Scrum -menetelmää päätettiin kokeilla sähköisen työpöydän pilottiprojektissa (Simola 2011). Ensisijaisena tutkimusongelmana on selvittää miten sähköisen työpöydän pilottiprojektissa mukana olleet asiakkaat ja muut asiantuntijat kokivat Scrumin käytön ja mitä kehitys- ja parannusehdotuksia tulee heidän mukaansa ottaa huomioon, jotta Scrumin käyttö olisi tulevissa Samlinkin

sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa. Samalla halutaan selvittää, mitä yritys ja asiakkaat ketterältä Scrumilta odottavat. Näkemyksistä on tarkoitus luokitella esiin nousseet teemat, ja sitä kautta määritellä tulokset. Ketterästä Scrum -menetelmästä selvitetään sen ideologia, sen käyttö sovelluskehitysprojektissa sekä sen ero perinteisiin menetelmiin, jotta saadaan selville, minkälaisia kysymyksiä tutkimukseen osallistuville henkilöille tulee esittää. Tutkimuskysymyksiä on kaksi:

1. Miten Scrumin pilottikäyttö onnistui Samlinkin sähköisen työpöydän pilottiprojektissa?
2. Mitä jatkossa pitäisi kehittää ja parantaa, jotta Scrumin käyttöönotto ja sen käyttö olisi Samlinkin tulevissa sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa?

1.4 Rajaukset

Opinnäytetyö koskee Samlinkin sovelluskehitysprojekteista vain sähköisen työpöydän pilottiprojektia, koska ketterä Scrum -menetelmä oli siinä pilottikäytössä. Projektissa on ollut mukana kaksi tiimiä, Espoon ja Jyväskylän tiimi. Opinnäytetyö koskee vain Espoon tiimiä, koska halutaan selvittää miten tiimi, joka ei ole toiminut ketterän Scrumin mukaisesti kokee sen käytön. Ketteristä menetelmistä keskitytään vain Scrumiin ja siinä Scrumin pääkohtiin.

1.5 Käsitteet ja niiden selitykset

Käsite	Selitys
Scrum	Ketterä sovelluskehitysmenetelmä
Product Backlog	Scrumin priorisoitu lista vaatimuksista, yksi Scrumin artefakteista
Sprint Backlog	Scrumin lista tehtävistä jotka toteutetaan Sprintin aikana, yksi Scrumin artefakteista
Product Owner	Yksi Scrumin rooleista; yhteistyössä Scrum tiimin ja asiakkaiden kanssa
Scrum Master	Yksi Scrumin rooleista; edistää tiimin tehokkuutta ja poistaa tiimin esteitä
Scrum-tiimi	Yksi Scrumin rooleista; vastaa kehitystyöstä
Daily Scrum Meeting	Yksi Scrumin palavereista; tapahtuu päivittäin ja kestää 15-minuuttia
Sprint Planning Meeting	Yksi Scrumin palavereista; tapahtuu aina ennen Sprintin alkua
Sprint	Sykli, jota toistetaan kunnes tuotos on valmis tai budjetti käytetty. Kestää suositusten mukaan 30 päivää
Sprint Review Meeting	Yksi Scrumin palavereista; mm. tuotoksen esittely

Sprint Retrospective Meeting	Yksi Scrumin palavereista; tiimi pohtii ja miettii miten se voisi parantaa työskentelyään ja prosessejaan
Este	Mikä tahansa asia tai ongelma joka häiritsee Scrum tiimin edistymistä

Taulukko 1. Käsitteet ja niiden selitykset

2 Teoreettinen tausta

2.1 Perinteinen sovelluskehitys ja perinteiset menetelmät

Perinteistä sovelluskehitystä tehdään tietyn ennalta määritellyn järjestyksen mukaisesti. Lisäksi suositaan roolipohjaisia tiimejä ja yksityiskohtaisia suunnitelmia jotka koskevat koko sovelluskehityksen elinkaarta. Kommunikointi tapahtuu pääasiassa dokumenttien kautta. (Chau, Maurer & Melnik 2002, 1).

Perinteisen sovelluskehityksen suurimpia ongelmia ovat olleet vaatimusten kerääminen ennen suunnittelua ja kehittämistä, koska usein vaatimuksia ei silloin vielä ymmärretä. Lisäksi vaatimukset ovat usein myös muuttuneet sovelluksen kehittämisen aikana (Sutherland 2001, 11; Chau ym. 2002, 3.)

Perinteisessä sovelluskehityksessä käytetyt menetelmät perustuvat olettamukseen, että koko järjestelmä voidaan määritellä ennen sen toteuttamista. Laajaa vaatimusten keräystä seuraa yksityiskohtainen suunnittelu ja näistä syntyvien määritysten avulla ohjataan toteutusta ja testausta. Kehittäjät erikoistuvat tiettyihin tehtäviin kuten suunnitteluun ja koodaukseen. Lisäksi projektihallinnallisia epävarmuuksia kuten aikataulun ja kustannusten ylittymistä yritetään vähentää tarkan suunnittelun, dokumentoinnin ja tiukan prosessin avulla (Balijepally, Mahapatra & Nerur 2006, 57).

Perinteiset sovelluskehitysmenetelmät perustuvat muun muassa vesiputousmalliin, spiraalimalliin tai niiden erilaisiin variaatioihin tai Rational Unified Process malliin.

2.1.1 Vesiputousmalli

Vesiputousmalli on yksi tunnetuimmista perinteisen sovelluskehityksen menetelmistä, ja sitä on käytetty laajasti sen ennustettavuuden, vakauden ja korkean varmuuden takia (Cho 2008, 189.) Vesiputousmalli on suoraviivainen malli, jossa panostetaan dokumentaatioon. Se etenee määriteltujen vaiheiden mukaisesti jolloin aloitus tapahtuu vaiheesta yksi ja se täytyy olla suoritettu ennen kuin voi siirtyä vaiheeseen kaksi jne. Jos jossakin vaiheessa tapahtuu virhe, täytyy palata aikaisempaa vaiheeseen ja tehdä vaadittavat askeleet uudestaan (Waterfall Model 2011). Vesiputousmallin klassisen version kehitti vuonna 1970 Royce ja vuonna 1976

Boehm muutteli sitä, jotta voitaisiin paremmin vastata sovelluskehitysprojektien monimutkaisuuteen (Davis, Bersoff & Comer 1988, 1453.)

Vesiputousmallin käyttö rohkaisi määrittelemään mitä sovelluksen tulisi tehdä sekä suunnittelemaan miten komponentit integroidaan ennen sovelluksen toteuttamista. Se antoi myös projektipäällikölle mahdollisuuden seurata edistymistä tarkemmin. Vesiputousmalli vaatii myös tiettyjen dokumenttien tekemisen, joiden avulla testataan ja ylläpidetään järjestelmää. Lisäksi vesiputousmalli vähentää kehitys- ja ylläpitokustannuksia kaikkien edelle mainittujen johdosta ja mahdollistaa sovellusta kehittävän organisaation olevan järjestelmällinen ja mukautuva. Vesiputousmalli kuvataan yleensä viisi - seitsemänvaiheisena mallina, ja vaiheiden nimissä saattaa olla eri variaatioita (Davis ym. 1988, 1453.)

2.1.2 Spiraalimalli

Spiraalimalli on kehittynyt vuosien aikana ja se perustuu kokemuksiin joita on saatu kun vesiputousmallia on sovellettu suurissa sovelluskehitysprojekteissa (Boehm 1988, 65.) Se on riskivetoinen prosessimalli, jonka ideana on sovelluksen toteuttaminen useammassa jaksossa vähentämällä samalla merkittävimpiä jäljellä olevia riskejä (Boehm & Hansen 2001, 4).

Jokainen jakso alkaa tunnistamalla tuotteeseen liittyviä tavoitteita kuten suorituskyky ja toiminnallisuus, toteuttamisen vaihtoehtoja kuten malli a, uudelleen käyttö, ostaminen sekä näiden vaihtoehtojen rajoituksia kuten kustannukset ja aikataulu. Seuraavassa vaiheessa arvioidaan tavoitteisiin, toteuttamisen vaihtoehtoihin ja niiden rajoituksiin liittyviä ongelmia. Tässä arvioinnissa löydetään yleensä myös epävarmuuksia, jotka saattavat muodostua projektin riskeiksi. Tällöin mahdolliset riskit täytyy ratkaista. Työkaluina voidaan käyttää esimerkiksi ohjelman protoja, simulaatioita tai vertailututkimusta. Tämän jälkeen ohjelma kehitetään ja lopuksi siitä syntyy versio joka testataan. Jokaiseen vaiheeseen kuuluu katselmointi, jonka tarkoituksena on varmistaa, että kaikki osallistujat ovat sitoutuneet seuraavaan vaiheeseen (Boehm 1988, 65).

2.1.3 Rational Unified Process

Rational Unified Prosessi eli RUP on sovellusprosessi, jonka tarkoituksena on ohjata organisaation sovelluskehitystä. RUP:ssa on säännöllinen ja suunniteltu prosessi rakenne ja sen kuvaus käyttää oliosuuntautuneista lähestymistapaa. Prosessia ei ole tarkoitus noudattaa sokeasti vaan siitä täytyy tehdä niin kevyt kuin mahdollista säilyttäen silti sen päämäärän auttaa kehittäjiä toteuttamaan nopeasti laadukkaita sovelluksia (Kruchten 2001, 1 - 3). Prosessin tarkoituksena on kertoa miten UML:ää käytetään tehokkaasti. UML on standardoitu kieli, jonka avulla voidaan selkeästi kommunikoida vaatimuksista, arkkitehtuurista ja malleista. Prosessia

voidaan muotoilla ja se sopii hyvin sekä pieniin että isoihin sovelluskehitystiimeihin (Rational 1998, 1).

Muita perinteisiä sovelluskehitysmenetelmiä ovat muun muassa vesiputousmallin ja spiraalimallin variaatiot sekä V-malli.

2.2 Ketterä sovelluskehitys

Useat sovelluskehitysprojektit myöhästyvät, ylittävät budjetin eikä valmis sovellus täytä asiakkaiden tarpeita ja vaatimuksia (The Agile Management Company 2010, 1.) Kilpailu on kasvanut ja liiketoimintaympäristöt ovat muuttuneet haastaviksi ja tämän takia tarvitaan nopeampaa ja kustannustehokkaampaa sovelluskehitystä (Ness Technologies 2010, 6.) Sovelluskehityksessä tarvitaan lyhyempiä kehityssyklejä ja korkeampaa näkyvyyttä ja asiakkaat eivät myöskään halua kohdata yllätyksiä. Kaikki tämä on johtanut siihen, että asiakkaat ovat alkaneet vaatia ketteryyttä ja organisaatioiden täytyy ottaa käyttöön kevyet ja ketterät sovelluskehitys-menetelmät tukeakseen ja vastatakseen asiakkaiden vaatimuksiin (Sureshchandra & Shrinivasavadhani 2008, 97).

Ketterän sovelluskehityksen avulla on saavutettu useampien tutkimusten, raporttien ja onnistuneiden projektien mukaan parempia tuloksia tuotteen nopeampaan saamiseen markkinoille, alempiin virhekustannuksiin, parempaan tiimihenkeen ja tyytyväisempiin asiakkaisiin (The Agile Management Company 2010, 1).

Tuotteen saaminen nopeammin markkinoille tuo lisäarvoa asiakkaille, koska he pääsevät käyttämään tuotetta aikaisemmin (Elssamadisy 2009, 4.) Nopeamman markkinoille pääsyn mahdollistavat ketterät tiimit, joiden työskentelyn tulokset nähdään projektien parempana laatuna. Työntekijät ovat vapaita tekemään laadukkaampaa työtä ja he näkevät työnsä tulokset nopeammin. Tämä kasvattaa yleensä tyytyväisyyttä. Työntekijät, jotka ovat tyytyväisempiä työhönsä ja työnantajaansa, ovat paremmin sitoutuneita tekemäänsä työhön. Myös laatu paranee, koska työskentely vakaassa tahdissa estää huolimattomuutta. Lisäksi laatu paranee myös monien käytäntöjen, kuten parikoodauksen, refaktoroinnin ja aikaisen ja automaattisen testauksen avulla (Cohn 2010, 10 - 14).

Myös kustannuksia voidaan säästää nopeammalla markkinoille menolla sekä paremmalla markkinalaadulla ja -arvolla. Kustannuksiin vaikuttavat kuitenkin myös sovelluskehitykseen liittyvät kulut, ja siksi organisaatioiden tulisikin miettiä käytäntöjä, joiden avulla voidaan pienentää sovelluskehitystyön kustannuksia. Näitä ovat henkilötyötunti, ylläpitotyötunti sekä laitteistojen ja sovelluslustojen kustannukset. Käytännöt, jotka vähentävät joidenkin edellisten tai niiden kaikkien kustannuksia uhraamatta laatua, vähentävät järjestelmän kokonaiskus-

tannuksia. Toinen tapa pienentää kustannuksia on vähemmän koodin kirjoittaminen. ”80/20 sääntö” kertoo, että arviolta 20 prosenttia tuotteen ominaisuuksista on käytössä 80 prosenttia ajasta. Käytännöt, jotka auttavat tiimiä rakentamaan tärkeimmän, vähentää huomattavasti tuotteen kustannuksia ja tuottaa liiketoiminta-arvoa asiakkaille (Elssamadissy 2009, 25).

Ketterä sovelluskehitys parantaa liiketoiminta-arvoja kuten kasvattaa tuottoja tuottamalla aikaisemmin ja tietyn väliajoin jotakin toiminnallisuutta, parantaa laatua elinkaaren läpi tapahtuvalla integraatiotestauksella. Lisää asiakkaiden tyytyväisyyttä, koska he ovat aktiivisesti koko ajan mukana sekä lieventää riskejä korkean näkyvyyden ja vaiheittaisten versioiden tuloksena (Ness Technologies 2010, 3).

Ketterän sovelluskehittämisen käyttöönotto edellyttää visiota, rohkeutta ja niiden liiketoiminnallisten ja organisatoristen etujen ymmärtämistä jotka liittyvät ketterään. Johtajilla, jotka kannattavat ketterää sovelluskehitystä, on voimakas halu nopeuttaa liiketoiminta-arvon tuottamista yrityksessään ja parantaa niitä tapoja joiden avulla tätä arvoa tuotetaan. Ketterä sovelluskehittäminen vaatii myös päästämään irti vanhasta ja omaksuma uusia toimintatapoja. Tuottavuuden ja laadun maksimoimiseksi osastojen väliset siilot erotetaan ja muodostetaan tiimejä. Tiimit jakavat yhdessä avoimen vision luottaen toisiinsa ja tuottaen mahdollisimman hyvää liiketoiminta-arvoa määrättyssä ajassa. Käyttöönotto vaatii myös johtajien tukea. Johdon täytyy kouluttautua, jotta he ymmärtävät ketterän sovelluskehityksen edut ja tulokset sekä siihen liittyvät haasteet. Odotuksien ketterältä sovelluskehitykseltä täytyy olla realistiset. Ketterää ei saa nähdä ”hopea luotina” vaan sen sijaan uutena, kurinalaisena sovelluskehityksen lähestymistapana jonka avulla pystytään paremmin hallitsemaan nopeita toimitusjaksoja ja muutoksia (The Agile Management Company 2010, 3 - 5).

Ketterän sovelluskehittämisen käyttöönotossa täytyy seurata seuraavia askelmia.

Täytyy olla aikaa ketterän käytäntöjen ymmärtämiseen ja tukea ketterän vaatimaa muutosta. Lisäksi täytyy huolehtia ketterän käyttöönotosta koko organisaatiossa ja sijoittaa taitoihin ja työkaluihin joiden avulla mahdollistetaan ketterän toimintatavan käyttö jopa suurimmissa sovelluskehitysprojekteissa (The Agile Management Company 2010, 2 - 3).

2.3 Ketterät sovelluskehitys -menetelmät

Ketteriä sovelluskehitysmenetelmiä on kehitetty 1990-luvun alusta lähtien kehittämään sovelluksia ja pelastamaan sovelluskehitysprojekteja epäonnistumasta (Cho 2008, 188.) Ne ovat nykyaikaisia lähestymistapoja sovelluksien kehittämiseen perustuen asiakkaiden yhteistyöhön, tiimityöskentelyyn, iteratiiviseen kehittämiseen ja muutoksiin vastaamiseen (Rico ym. 2009, 1.) Lyhyen sovelluskehityselinkaaren sekä iteratiivisen ja inkrementaalisen prosessin vuoksi,

ketteriä menetelmiä on käytetty paljon liiketoimintasektoreilla, joissa vaatimukset ovat suhteellisen epävakaita (Cho 2008, 188.)

Ketterissä sovelluskehitysmenetelmissä painotetaan asiakasyhteistyötä sopimusneuvottelujen sijaan, yksilöitä ja vuorovaikutusta prosessien ja työkalujen sijaan, toimivaa sovellusta dokumentaation sijaan sekä muutoksiin vastaamista suunnitelman noudattamisen sijaan. Nämä ovat ketterien menetelmien merkittävimpiä arvoja (Rico, Sayani & Sone 2009, 7).

Asiakkaiden yhteistyö sopimusneuvottelujen sijaan tarkoittaa asiakkaiden kuuntelemista. Asiakkailta kysytään mitä he haluavat ja näiden avulla varmistetaan heidän tarpeensa. Liiketoiminta-arvoa syntyy, kun kysytään asiakkailta mitä he haluavat ja toteutetaan se. Yksilöt ja vuorovaikutus prosessien ja työkalujen sijaan tarkoittaa, että kehittäjillä on valtuus muodostaa tiimejä ja johtaa itseään. Tämän avulla saavutetaan rikasta kommunikointia kehittäjien välillä, asiantuntevia kehittäjiä ja mikä tärkeintä, kehittäjien yhteistyöskentelyä. Toimiva ohjelmisto kattavan dokumentaation sijaan tarkoittaa, että toimivan sovelluksen tuottaminen on yksi korkeimmista prioriteeteista ketterissä menetelmissä. Kaikki resurssit on keskitetty rakentamaan toimivaa sovellusta kahdesta neljään viikon jaksoissa. Tämä tunnetaan myös iteratiivisena ja inkrementaalisenä kehityksenä. Asiakas maksaa toimivasta sovelluksesta, ei dokumentaatiosta ja siksi liiketoiminta-arvon maksimoimiseksi luodaan toimivia sovelluksia. Muutoksiin vastaaminen suunnitelman noudattamisen sijaan tarkoittaa joustavaa suhtautumista muutoksiin eli ymmärretään asiakkaiden tarpeet, kehitetään sovelluksia vaiheittain, näytetään tuotos asiakkaille ja toistetaan tätä iteraatiota, kunnes asiakas on tyytyväinen (Rico ym. 2009, 8 - 10.)

Ketteriä sovelluskehitysmenetelmiä on muun muassa extreme programming (XP), adaptive software development (ASD), agile modeling (AM) ja Scrum.

2.3.1 Extreme programming (XP)

Extreme programming (XP) on kevyt, tehokas, joustava ja ennakoitavissa oleva sovelluskehitysmenetelmä. Se lupaa vähentää projektien riskejä sekä parantaa tuottavuutta ja muutoksiin vastaamista (Beck 1999, 8 - 9). XP:n päämääränä on onnistunut sovelluskehitys huolimatta epävarmuuksista ja jatkuvasti muuttuvista vaatimuksista. Sen joitakin pääpiirteitä on lyhyet iteraatiot ja pienet versiot sekä nopea palaute, läheinen yhteistyö asiakkaiden kanssa sekä jatkuva kommunikointi ja yhteistyö (Abrahamsson, Warsta, Siponen & Ronkainen 2003, 2).

2.3.2 Adaptive software development (ASD)

Adaptive software development (ASD) kehitettiin projekteihin joissa vallitsee nopeus, muutokset ja epävarmuus. Siinä käytetään dynaamista elinkaarta, joka omistautuu jatkuvaan op-

pimiseen ja suuntautuu jatkuviin muutoksiin, uudelleen arviointiin, tulevaisuuden epävarmuuksiin sekä intensiiviseen yhteistyöhön kehittäjien, testaajien ja asiakkaiden välillä (Highsmith 2000, 23).

2.3.3 Agile modeling (AM)

Agile modeling (AM) on sovelluskehitysmenetelmä tehokkaaseen mallintamiseen ja dokumentointiin. Se perustuu mallintamisperiaatteisiin kuten yksinkertaisuuteen, muutoksien tukemiseen ja nopeaan palautteeseen (Ambler 2006, 1). Sen pohjimmainen idea on rohkaista kehittäjiä tuottamaan riittävän kehittyneitä malleja tukemaan mallitarvetta ja dokumentaatiota pitäen silti mallien ja dokumentaatioiden määrän niin vähäisenä kuin mahdollista (Abrahamsson ym. 2003, 2).

Scrumista kerrotaan tässä tutkimuksessa.

Muita ketteriä sovelluskehitysmenetelmiä ovat crystal family of methodologies, dynamic system development method, feature-driven development ja Internet-speed development (Abrahamsson ym. 2003, 2.)

2.4 Perinteiset menetelmät vastaan ketterät menetelmät

Vesiputousmallin tyylisiä sovelluskehitysmenetelmiä on käytetty paljon suurissa projekteissa niiden ennustettavuuden, vakauden ja korkean varmuuden takia. Niissä on kuitenkin monia puutteita, kuten nopeasti muuttuvien vaatimusten hidas mukauttaminen, taipumus budjetin ja aikataulujen ylittymiseen sekä se, että tuotetaan vähemmän piirteitä ja toiminnallisuutta kuin mitä oli määritelty. Ne ovat myös epäonnistuneet tuottamaan parannusta tuottavuuteen, luotettavuuteen ja yksinkertaisuuteen. Edellisten takia niitä kutsutaan yleensä raskaan sarjan kehitysmenetelmiksi (Cho 2008, 188 - 189).

Ketterien sovelluskehitysmenetelmien ominaispiirteitä ovat muun muassa vakuuttaa asiakkaat aikaisen ja jatkuvan sovelluksen tuottamisella, vaatimuksien muutoksien tukeminen, päivittäinen työskentely liiketoiminta ihmisten kanssa, ihmisiin luottaminen, sovelluksen käyttäminen ensisijaisena edistyksen mittarina, vakaan kehityksen edistäminen, teknisen erinomaisuuden ja hyvän suunnittelun huomioiminen, yksinkertaisuuden ylläpitäminen sekä itseorganisoi-
tuvien tiimien edistäminen (Cho 2008, 190).

Perinteisissä menetelmissä järjestelmäkehitystä ohjaa elinkaarimalli, kuten vesiputous- tai spiraalimalli tai jokin näiden variaatio. Elinkaarimalli määrittelee suoritettavat tehtävät ja roolit, jotka suorittavat nämä tehtävät sekä vaiheiden halutut tuotokset. Nämä menetelmät tuottavat myös suuren määrän dokumentaatiota, ja kommunikaatio projektin osallistujien

kesken tapahtuu näiden dokumenttien avulla. Asiakkailta on tärkeä rooli määrittelyvaiheessa, mutta muuten heidän osallistumisensa on hyvin vähäistä. Ketterät menetelmät taas käsittelevät arvaamatonta luottamalla ihmisiin ja heidän luovuuteensa enemmän kuin prosessiin. Ketterille menetelmille on tunnusomaista lyhyet kehysitieraatiot, yhteistyössä tehdyt päätökset sekä nopeat palautteet ja muutokset (Nerur ym. 2005, 75).

Nerur ym. (2005, 75) kuvaavat perinteisen ja ketterän sovelluskehityksen eroja seuraavasti

	Perinteinen	Ketterä
Perusolettamukset	Järjestelmät ovat täysin yksityiskohtaisia, rakennetaan huolellisen ja kattavan suunnittelun avulla	Korkealaatu, joustava ohjelmisto voidaan kehittää pienillä tiimillä, nopea palaute ja muutokset
Hallinta	Prosessilähtöinen	Ihmislähtöinen
Johtamistyyli	Määrääminen ja hallinta	Johtaminen ja yhteistyö
Tiedonhallinta	Suora	Epäsuora
Rooli tehtävät	Yksilö - suosii erikoistumista	Itseorganisoituvat tiimit - rohkaisee roolin vaihtoihin
Kommunikointi	Muodollista	Epämuodollista
Asiakkaiden rooli	Tärkeä	Kriittinen
Projektin sykli	Tehtävät ja aktiviteetit ohjataan	Tuotteen piirteet ohjaavat
Kehittämismalli	Elinkaarimalli (Vesiputous, Spiraaali tai joku variaatio)	Kehitystoimitusmalli
Haluttu Organisaatiolinen muoto/rakenne	Mekanistinen (byrokraattinen)	Orgaaninen (joustava ja osallistava, rohkaisee yhteistyöhön)
Teknologia	Ei rajoitteita	Suosii oliopohjaista teknologiaa

Taulukko 2. perinteinen vastaan ketterä sovelluskehitys (Nerur ym. 2005, 75.)

Asiakkaat eivät yleensä tiedä mitkä ovat heidän oikeat tarpeensa ja halunsa ennen kuin he näkevät toiminnallisen osan järjestelmästä (Sutherland 2001, 11; Chau ym. 2002, 3.) Tämän vuoksi monien ketterien -menetelmien mukaan tehdään pieniä ja säännöllisiä versioita, joiden avulla kehittäjät ja asiakkaat näkevät sovelluksen tietyn väliajoin ja ymmärtävät paremmin sen toimintaa. Lisäksi pienet ja säännölliset versiot antavat asiakkaille mahdollisuuden määrittellä heille sopivia vaatimuksia. Asiakkaat myös odottavat, että kehittäjät pystyvät oikean tiedon avulla ratkaisemaan ongelmia tehokkaasti ja nopeasti. Edellisten avulla vältetään myös tyypillinen perinteisten menetelmien tilanne, jossa asiakkaat nähtyään suuren määrän piirteitä ilmoittavat lopulta, että toteutetut piirteet eivät tyydytä heidän tarpeitaan. Näin tapahtuu yleensä joko siksi, että kehittäjät ja asiakkaat eivät ole tehneet yhteistyötä tai yksinkertaisesti siksi, että asiakkaiden tarpeet ja vaatimukset ovat muuttuneet siitä, kun vaatimukset kerättiin ja jäädytettiin. (Chau ym. 2002, 3).

Perinteisissä menetelmissä projektinhallinnan lähestymistapana on määräävä ja hallitseva johtamistapa. Niissä luodaan projektisuunnitelma, joka listaa kaikki tiedossa olevat tehtävät. Projektipäällikön työnä on toteuttaa tämä suunnitelma ja lisäksi projektipäällikkö johtaa eli

ohjaa, hallitsee ja valvoo (Cohn & Schwaber 2003, 4). Projektipäälliköllä menee myös joka viikko useita päiviä ylläpitäessään aikatauluja, tarkastellessaan kaavioita, antaessaan tehtäviä tiimin jäsenille ja seurattessaan tehtäviä (The Agile Management Company 2010, 5.)

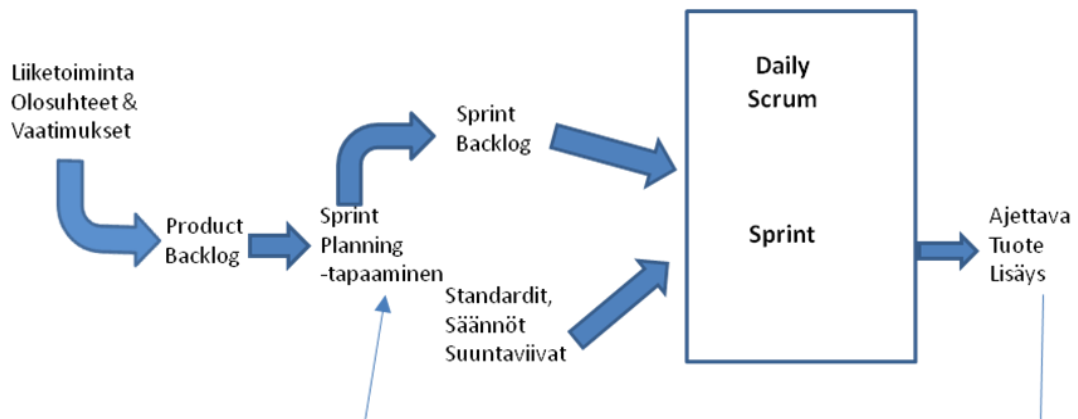
Ketterissä menetelmissä projektinhallinnan lähestymistapana painotetaan enemmän johtamista kuin hallintaa. Siinä ei luoda yksityiskohtaista suunnitelmaa, joka kuvaa kaikkien toimintojen järjestystä, vaan projektipäällikkö työskentelee yhdessä asiakkaiden kanssa laatiakseen yleisen joukon vaatimuksia, joissa kasvua tai muutoksia voi tapahtua (Cohn & Schwaber 2003, 4). Projektipäällikölle jää myös enemmän aikaa muun muassa projektien esteiden poistamiseen, yhteistyöhön asiakkaiden kanssa sekä tavoitteiden kertomiseen ja niiden edistämiseen, koska ketterissä menetelmissä tiimit ovat enemmän itseorganisoituneita ja huolehtivat itse omista tehtävistään (The Agile Management Company 2010, 5).

Siirtyminen vesiputousmallista ketterään malliin ei tapahdu yhdessä yössä tai yksittäisenä askeleena. Vie aikaa ennen kuin ihmiset unohtavat vanhat perinteiset käytännöt ja siirtyvät kohti ketteriä käytäntöjä. Ketterien käytäntöjen muutoksen edistämiseksi täytyy olla kärsivällinen, positiivinen ja sinnikäs. Ketterät käytännöt edellyttävät ajattelutavan muuttamista. Jos jotkut henkilöt ovat jäykkiä ja vastustavat muutosta, on parasta, että nämä henkilöt eivät ole mukana ketterissä projekteissa. Muuttaminen täytyy tehdä asteittain ja huolellisesti (Sureschandra & Shrinivasavadhani 2008, 101).

2.5 Ketterä Scrum -menetelmä

Scrumin kehittivät ensimmäiseksi Jeff Sutherland ja muut Easel-yhtiöstä vuonna 1993, ja vuonna 1995 Ken Schwaber ja Jeff Sutherland kehittivät Scrumista muodollisen kuvauksen. Scrum luotiin kahdesta syystä: ”(1) olemassa olevat menetelmät eivät yksinkertaisesti toimineet ja (2) uutta menetelmää tarvittiin varmistamaan projektien onnistuminen” (Rico ym. 2009, 25 - 26). Scrumin lähtökohtana on, että sovelluskehitys on liian monimutkaista ja arvaamatonta, jotta se voitaisiin suunnitella tarkasti etukäteen. Päinvastoin, sovelluskehityksen täytyy varmistaa näkyvyys, tarkastelu ja mukauttaminen. Erilaiset ympäristö- ja tekniset tekijät täytyy pystyä hallitsemaan pysyvästi, jotta voidaan mukauttaa muutoksia joustavasti. Tämä saavutetaan toistuvalla ja vähittäisellä kehitysprosessilla (Mahnic & Drnovscek 2005, 1).

Scrumin etuja ovat ohjelmistojen aikaisempi tuottaminen ja niiden korkeampi laatu (Sutherland 2003, 3; Cohn 2010, 11). Sen etuja ovat myös työntekijöiden parempi sitoutuminen ja tyytyväisyyden parantuminen, tuotteen nopeampi saaminen markkinoille sekä asiakkaiden tyytyväisyyden lisääntyminen (Cohn 2010, 11.) Lisäksi piirteet pystytään esittämään asiakkaalle aikaisin ja usein ja näin ollen asiakkaat voivat koko ajan tarkkailla kehitystyön edistymistä ja priorisoida muutoksia Sutherland (2003, 3.)



Kuva 1. Scrum prosessi (Schwaber & Beedle 2002, 8.)

Scrumissa kaikki työt tehdään iteraatioissa eli Sprinteissä. Jokainen Sprintti kestää 30 peräkkäistä kalenteripäivää. Sprintti alkaa Sprint Planning -tapaamisella missä Product Owner ja tiimi yhteistyössä miettivät mitä on tarkoitus tehdä seuraavassa Sprintissä (Mahnic & Drnovscek 2005, 2). Product Backlogista valitaan korkeimman prioriteetin asioita ja tätä kautta Product Owner kertoo tiimille, mitä sen toivotaan toteuttavan ja tiimi kertoo Product Ownerille kuinka paljon toivotuista asioista se pystyy toteuttamaan toiminnallisuudeksi seuraavan Sprintin aikana (Mahnic & Drnovscek 2005, 2; Schwaber & Beedle 2002, 7.)

Product Backlog on priorisoitu lista, joka kuvaa kaikki ne asiat, jotka järjestelmän täytyy sisältää kuten toiminnallisuus, piirteet ja teknologia. Listan sisältö voi tulla mistä vain: käyttäjiltä, asiakkailta, myynniltä, markkinoinnista, asiakaspalvelusta tai kehittäjiltä. Kuitenkin vain Product Owner voi priorisoida listaa ja hän päättää siitä, missä järjestyksessä asiat Product Backlogissa ovat (Schwaber & Beedle 2002, 7). Priorisoitu Product Backlog luodaan yleensä yhdessä asiakkaan kanssa ja heidän kanssaan myös sovitaan, mitä toimintoja lähdetään kehittämään (Schwaber & Beedle 2002, 21.)

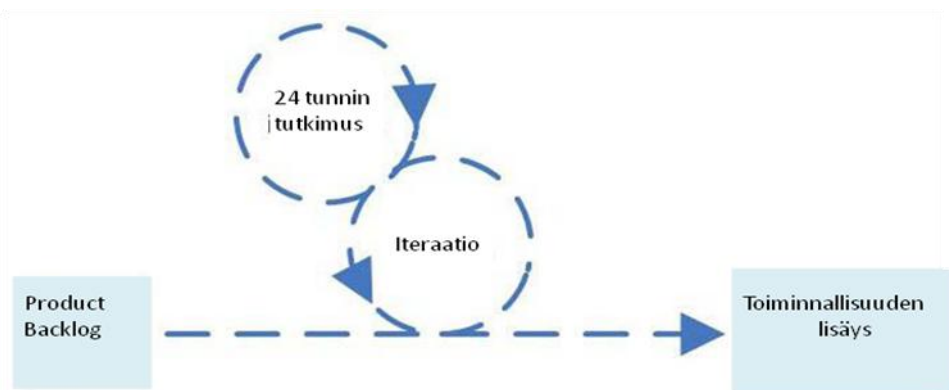
Sen jälkeen kun on päätetty, mitä seuraavassa Sprintissä tehdään, tiimi luo Sprint Backlogin eli listan tehtävistä, jotka täytyy kyseisen Sprintin aikana suorittaa (Mahnic & Drnovscek 2005, 2; Schwaber & Beedle 2002, 9.) Listan tehtävät kasvavat sitä mukaa kuin Sprintti kehittyy. Sprint Backlog pitäisi myös jakaa siten, että jokaisen tehtävän valmistuminen kestää noin 4-16 tuntia (Mahnic & Drnovscek 2005, 2).

Tiimi tapaa toisensa päivittäin 15 minuuttia kestävässä Daily Scrumissa (Mahnic & Drnovscek 2005, 2; Schwaber & Beedle 2002, 21.) Palaverin tarkoituksena on synkronoida kaikkien tiimin jäsenten työt ja aikatauluttaa palaverit, joita tiimi tarvitsee päästäkseen eteenpäin (Mahnic & Drnovscek 2005, 2.) Palaverissa tarkistetaan edistyksen tilanne ja havaitaan mahdolliset

esteet. Daily Scrum on erittäin hyvä paikka havainnollistamaan, kuinka paljon edistystä tiimi on saanut aikaan (Schwaber & Beedle 2002, 9).

Jokaisen Sprintin lopuksi pidetään Sprint Review -tapaaminen missä esitellään tiimin tekemät lisäykset tuotteeseen (Mahnic & Drnovscek 2005, 2 - 3; Schwaber & Beedle 2002, 21.) Tiimi joko toteutti sen mitä oli suunniteltu, käytti toteutettua muuten hyödyksi tai ei voinut hyödyntää toteutettua mitenkään (Schwaber & Beedle 2002, 10).

Seuraavassa kuvassa on Mahnicin ja Drnovscekin (2005, 2) kuvaama Scrumin runko. Alimmainen ympyrä esittää kehitystoiminnan iteraatioita, joita tapahtuu yksi toisensa jälkeen. Jokaisen iteraation tulos on lisäys tuotteeseen. Ylempi ympyrä esittää päivittäistä tutkimusta, joka tapahtuu iteraation aikana ja jossa tiimin jäsenet tutkivat toistensa toimintoja ja tekevät tarkoituksenmukaisia muokkauksia. Iteraation panee liikkeelle lista vaatimuksia. Tämä sykli toistuu kunnes projekti on valmis.



Kuva 2. Scrumin runko (Mahnic & Drnovscek 2005, 2.)

Scrum toteuttaa tätä iteratiivista inkrementaalista runkoa läpi kolmen roolin: Product Ownerin, Scrum-tiimin ja Scrum Masterin.

- Product Ownerin vastuulla on edustaa kaikkien osapuolien näkemyksiä. Hän myös ylläpitää Product Backlogia.
- Tiimin vastuulla on kehittää toiminnallisuutta. Tiimit ovat itseohjautuvia ja niiden vastuulla on selvittää miten Product Backlogista saadaan tuotettua lisäyksiä tuotokseen iteraation aikana. Tiimin jäsenet ovat yhdessä vastuussa jokaisesta iteraatiosta.
- Scrum Masterin vastuulla on Scrum prosessin johtaminen eli Scrumin opettaminen projektissa mukana oleville, Scrumin toteuttaminen siten, että se sopii organisaation kulttuuriin ja silti tuottaa odotetut hyödyt ja lisäksi Scrum Masterin täytyy varmistaa, että kaikki noudattavat Scrumin sääntöjä ja käytäntöjä (Mahnic & Drnovscek 2005, 2).

2.6 Ketterän Scrum-menetelmän käytännöt ja niiden käyttö sovelluskehityksessä

Schwaber & Beedle (2002, 31) suosittelevat Scrum -käytäntöjen tiukkaa noudattamista kunnes saavutetaan kokemusta ja ymmärrystä siitä, miksi ja miten Scrum toimii. Kun Scrum toimii hyvin organisaatiossa ja ihmiset ovat omaksuneet ne arvot, joiden mukaan Scrum toimii, voidaan tehdä muutoksia ja soveltaa Scrumia. Schwaberin ja Beedlen mukaan Scrumin käyttö täytyy opetella käytännössä eikä vain kirjoja lukemalla.

Monet ketterät käytännöt saattavat alussa tuntua tuskallisilta ja siksi Elssamadisy (2009, 231) suosittelee käyttämään ketteränmenetelmän mentoria. Myös Agile Management Company (2010, 4) suosittelee mentorin käyttöä. Heidän mukaansa useimmissa tapauksissa on suositeltavaa käyttää ulkopuolista mentoria auttamaan muutoksessa, koska sisäiset mentorit ovat usein vähemmän objektiivisia. Myös Ness Technologies (2010, 6) suosittelee ulkoisen mentorin käyttöä. Ness Technologiesin mukaan mentorilla täytyy olla omakohtaista kokemusta kaikista ketteristä käytännöistä, työkaluista ja tekniikoista ja lisäksi mentorin täytyy pystyä jakamaan kokemuksensa tiimien kanssa.

Seuraavassa esitellään Scrumin käytäntöjä sekä sitä miten niitä käytetään sovelluskehityksessä. Scrumiin kuuluu Daily Scrum, Sprint Planning, Sprint Review ja Retrospective -tapaamiset, Product Backlog ja Sprint Backlog -artefaktit sekä itse Sprintti (Settler 2011, 88- 108; Schwaber & Beedle 2002, 31 - 56; Mahnic & Drnovscek 2005, 2).

2.6.1 Sprint Planning Meeting

Sprint Planning -tapaamisessa Product Owner ja Scrum-tiimi määrittelevät seuraavan Sprintin tavoitteet ja toiminnallisuudet valitsemalla asioita Product Backlogista (Cho ym. 2008, 191.) Kun Product Backlogista on valittu asiat seuraavaan Sprinttiin, Sprintin tavoitteet on määriteltävä. Tämän jälkeen tiimi määrittelee, mitä työtä täytyy tehdä, jotta tavoitteet saavutetaan. Jokaisen tiimin jäsenen täytyy olla määrittelyssä paikalla. Tiimi voi kutsua määrittelyyn myös muita tarvittavia henkilöitä. Product Owner on usein myös mukana. Jos johto ja käyttäjät ovat tapaamisessa mukana, eivät he saa tehdä tai sanoa mitään sellaista, joka aiheuttaa sen, että tiimi ei tee itse valintojaan ja päätöksiään. (Schwaber & Beedle 2002, 48).

Tiimi muodostaa listan niistä tehtävistä joita sen täytyy suorittaa saavuttaakseen Sprintin tavoitteet. Tätä listaa kutsutaan Sprint Backlogiksi. Sprint Backlogin pitää sisältää tarpeeksi yksityiskohtia jotta tiimi pystyy pilkkomaan tehtävät tarpeeksi pieniksi. Tehtävien pilkkomisen jälkeen tiimi arvioi tehtävien työmäärät. Tehtävistä yritetään tehdä niin pieniä, että yhden tehtävän tekemiseen menee 4-16 tuntia (Cho ym. 2008, 192). Tiimi muuttaa Sprint Backlogia Sprintin aikana. Kun se alkaa muodostua yksittäisiksi tehtäviksi, voidaan huomata, että tehtäviä tarvitaan lisää tai vähemmän tai, että annettu tehtävä vie enemmän tai vähemmän

aikaa kuin oli odotettu. Jos uutta työtä vaaditaan, tiimi lisää sen Sprint Backlogiin. Kun tehtävät ovat työn alla tai valmiita, arvio jäljellä olevasta työstä ylläpidetään. Kun tehtävät näyttävät tarpeettomilta, ne poistetaan. Sprint Backlog on hyvin näkyvä, reaaliaikainen kuva työstä, jonka tiimi suunnittelee tekevänsä Sprintin aikana, ja se kuuluu ainoastaan tiimille (Schwaber & Beedle 2002, 50).

Joskus Scrum-tiimi huomaa, että se on valinnut liikaa tehtäviä Product Backlogista tehtäväksi yhden Sprintin aikana. Jos näin tapahtuu, Scrum Master tapaa välittömästi Product Ownerin ja Scrum-tiimin (Settler 2011, 107). Yhteisymmärryksessä tunnustetaan, mitä voidaan poistaa, jotta Sprintin tavoitteet on silti mahdollista saavuttaa. Jos mitään ei voida poistaa, tunnustetaan toiminnallisuuksia, joiden laajuutta tai syvyyttä voidaan vähentää. Tiimin taito Sprintin suunnittelussa paranee kolmannen tai neljännen Sprintin jälkeen. Kun tiimi oppii tuntemaan Scrumin prosesseja paremmin ja alkaa ymmärtää toiminnallisuutta ja teknologiaa tiimi sitoutuu työhön paremmin ja pystyy tekemään enemmän tehtäviä (Schwaber & Beedle 2002, 50).

2.6.2 Product Backlog

Product Backlog on priorisoitu lista vaatimuksista (Settler 2011, 88; Elssamadisy 2009, 81; Schwaber & Beedle 2002, 33; Cohn 2010, 235.) Se listaa kaikki vaatimukset, joita tarvitaan järjestelmään, projektiin ja tuotteeseen (Settler 2011, 88.) ja mahdollistaa vaatimusten muuttamisen saavuttaen silti määräajan ja tuottaen silti toiminnallisuuden arvokkaimmat piirteet (Elssamadisy 2009, 81.)

Product Backlog on aluksi keskeneräinen lista, joka kuvaa asiat, joita tuote tai järjestelmä tarvitsee. Product Backlog kehittyy tästä ensimmäisestä listasta sitä mukaan kuin asiakkaat ymmärtävät paremmin, mitä he haluavat. Product Backlogeja tulisi olla vain yksi, koska muuten tiimi ei tiedä mitä listaa sen pitäisi käyttää (Schwaber & Beedle 2002, 33 - 34). Product Backlogia ylläpitää ja sen priorisointijärjestyksestä vastaa Product Owner (Cohn 2010, 235; Schwaber & Beedle 2002, 34.) Product Owner myös varmistaa, että Product Backlog on näkyvä kaikille eli kaikki tietävät, millä asioilla on korkein prioriteetti. Scrumin mukaan on tärkeää, että vain yksi henkilö ylläpitää ja hallitsee Product Backlogin sisältöä ja prioriteetteja (Schwaber & Beedle 2002, 34).

Product Backlogin asioiden, jotka ovat tulossa pian tehtäviksi, täytyy olla riittävän yksityiskohtaisia, jotta ne voidaan toteuttaa, testata ja integroida yhteen Sprinttiin. Käyttäjätarinoiden, jotka ovat korkeimmalla Product Backlogissa, täytyy olla pieniä mutta järkevästi ymmärrettävissä. Käyttäjätarinat, jotka ovat alempana, ovat isompia ja niissä on vähemmän yksityiskohtia ja yleensä ne sisältävät vain sen verran tietoa, että ne voidaan suunnilleen arvioida ja sitten priorisoida (Cohn 2010, 243).

Kun Product Backlog on luotu, Product Owner keskustelee tarvittavien henkilöiden kanssa saadakseen arvion toteutuksen kestosta. Tämä arvio sisältää vaaditun arkkitehtuurin, suunnittelun, tekemisen ja testauksen suorittamisen arviot. Arviot ovat niin oikeita kuin sillä hetkellä pystytään antamaan. Arviointi on iteratiivinen prosessi. Se muuttuu sitä mukaa kuin saadaan enemmän tietoa Product Backlogin asioista ja niitä ymmärretään paremmin. Jos Product Owner ei saa selkeää ja uskottavaa arviota korkean prioriteetin Product Backlogista, hänen kannattaa miettiä joko asioiden uudelleen määrittämistä, prioriteetin alentamista tai asian siirtämistä epäselviin asioihin (Schwaber & Beedle 2002, 35).

Product Backlog arvio ei ole velvoite Scrum-tiimille eli arvio ei tarkoita sitä, että tietyn toiminnallisuuden toteuttamiseen on varattu aikaa juuri arvion verran eikä yhtään enempää. Arvio on lähtöpiste, paras arvio, jotta Sprintti voidaan kokemukseräisesti laatia ja hoitaa. Scrum-tiimi valitsee sen määrän Product Backlogista, jonka se kuvittelee pystyvänsä tekemään Sprintissä perustuen näihin arvioihin. Arviointi aloitetaan Product Backlogin korkeimman prioriteetin asioista (Schwaber & Beedle 2002, 35).

2.6.3 Sprintti

Sprintti on 30 kalenteripäivää kestävä ajanjakso, jona aikana tiimi työskentelee tuottaakseen sovitut lisäykset tuotteeseen ja ratkoakseen mahdolliset esille tulevat ongelmat. Scrum-tiimi on päättänyt, mitä se pyrkii saavuttamaan Sprintin aikana. Tiimin tarkoituksena on tehdä parhaansa, jotta monimutkaiset vaatimukset ja odottamaton teknologia saadaan lisättyä tuotteeseen (Schwaber & Beedle 2002, 50 - 52).

Ensimmäisellä Sprintillä on kaksi tarkoitusta. Ensin tiimin täytyy pystyttää kehitysympäristö, jossa se voi rakentaa toiminnallisuutta. Toiseksi tiimi rakentaa osan järjestelmää, joka esitellään asiakkaille 30 päivän kuluessa. Toiminnallisuuden esittäminen asiakkaille näin nopeasti edellyttää asiakkailta osallistumista ja miettimistä. Asiakkaat ymmärtävät, että heidän täytyy miettiä, mitä he järjestelmältä haluavat. Ensimmäinen Sprintti saa tiimin ja asiakkaat säännölliseen kolmenkymmenen päivän määrittely ja toimitus rytmiin (Schwaber & Beedle 2002, 58).

Tiimin työskennellessä ensimmäisessä Sprintissä, Product Owner ja asiakkaat lisäävät asioita, priorisoivat asioita tai tekevät asioihin muutoksia Product Backlogiin. Product Backlogin ei tarvitse olla valmis, vaan sen tulee sisältää tarpeeksi paljon korkean prioriteetin asioita jotta seuraavat muutamat Sprintit voidaan toteuttaa. Kun Product Owner ja asiakkaat saavat tuntea Scrumista, he pystyvät näkemään ja suunnittelemaan Product Backlogin asioita pidemmälle. Jos järjestelmä- tai tuotevisiota ei ole saatavilla, Product Owner ja asiakkaat ke-

hittävät sellaisen. Sen jälkeen he rakentavat Product Backlogin vision perusteella (Schwaber & Beedle 2002, 58).

Jokaisen Sprintin vaatimukset ovat ensisijaisesti asiakkaiden priorisoimia. Sen vuoksi vaatimukset, joilla asiakkaiden mielestä on korkein liiketoiminta-arvo, sisällytetään ensimmäiseen iteraatioon. Jäljelle jääneet vaatimukset arvioidaan uudelleen ja priorisoidaan seuraavia iteraatioita varten (Vinekar, Slinkman & Sridhar 2006, 32).

Testauksen kannalta on tärkeää, että Sprinttiin ei oteta liikaa isoja asioita Product Backlogista, koska tällöin testaamaan päästään vasta Sprintin loppuvaiheessa. Cohn suosittelee ottamaan yhden tai kaksi isompaa, ja sen lisäksi kaksi tai kolme pienempää asiaa. Tällöin osa kehittäjistä voi työskennellä suurien asioiden parissa ja antaa ne testattavaksi aina kun mahdollista. Muut kehittäjät voivat työskennellä pienempien asioiden parissa taatakseen testaajalle jonkinlaisen testausmahdollisuuden aikaisessa vaiheessa Sprinttiä (Cohn 2010, 208).

Sprintit vaikuttavat kaikkiin liiketoiminta-arvoihin. Ne tehostavat oppimista sallien tiimin näkevän määräajoin saavutetun edistyksen. Ihmiset oppivat yrityksen ja erehdyksen kautta kunhan heillä on aikaa pohtia virheitä. Edistyksen läpinäkyvyys on erittäin tärkeää oikean suunnittelun kannalta. Säännölliset ja pienet toteutukset ovat hyviä tiimin motivaation kannalta ja auttavat tiimiä pysymään tarmokkaina ja sitoutuneina. Säännöllinen rytmi auttaa tiimiä synkronoimaan työnsä ja auttaa tiimiä luottamaan siihen, että jatkuvaa edistymistä tapahtuu (Elsamadis 2009, 65 - 66).

Tiimi on sitoutunut tavoitteisiinsa ja ottanut vastuun siitä, että se rakentaa tarvittavat tuotelisäykset, jotta tavoitteet saavutetaan. Sillä on valta käyttäytyä niin kuin se parhaaksi näkee. Kukaan ulkopuolinen ei voi muuttaa sen työn laajuutta tai luonnetta, jota tiimi Sprintissä tekee eikä kenelläkään ole oikeutta lisätä Sprinttiin toiminnallisuutta tai teknologiaa. Organisaation johto on sijoittanut 30 päivää tiimiin Sprintissä. Välittämättä siitä mitä tiimi saavuttaa, se on omaksunut arvokasta tietoa vaatimuksista ja teknologiasta. Vaikka tiimi ei tuottaisi mitään todellista, se on joka tapauksessa käynyt läpi hyvin käytännöllisen oppimisprosessin (Schwaber & Beedle 2002, 51).

Sprintti voidaan keskeyttää ennen 30 päivän määräajan umpeutumista. Johto voi keskeyttää Sprintin jos se ei enää vastaa tarkoitustaan. Tarkoitus voi muuttua esimerkiksi siksi, että Sprintin tavoitteet ovat tulleet turhiksi, markkinatilat tai teknologiset vaatimukset ovat muuttuneet tai johto on jostain syystä muuttanut mielensä. Kuitenkin, koska Sprintin pituus on niin lyhyt, ei johdon ole yleensä järkevää keskeyttää Sprinttiä (Schwaber & Beedle 2002, 53).

Joskus tiimi itse voi päättää, että Sprintti pitäisi keskeyttää ja tällöin sillä täytyy olla myös oikeus siihen. Tiimille muodostuu Sprintin aikana parempi käsitys omasta osaamisestaan ja projektin vaatimuksista, jolloin tiimi voi Sprintin puolivälissä huomata, ettei se pysty saavuttamaan Sprintin tavoitteita. Vaikka tiimin käsitys tehtävästä työstä ei olisikaan muuttunut, silti voi olla tarpeen keskeyttää Sprintti. (Schwaber & Beedle 2002, 53).

2.6.4 Daily Scrum Meeting

Daily Scrum - tapaaminen on 15 minuuttia kestävä tapaaminen, jonka tarkoituksena on varmistaa, että jokainen kehitystiimin jäsen tietää, missä toteutuksessa ollaan menossa (Settler 2011, 90). Daily Scrum - tapaamisessa kysytään jokaiselta tiimin jäseneltä 3 kysymystä:

1. Mitä olet tehnyt viime palaverin jälkeen?
2. Mitä aiot tehdä ennen huomista Daily Scrumia?
3. Estääkö joku tai jokin sinua tekemästä työtä niin tehokkaasti kuin mahdollista?

(Schwaber & Beedle 2002, 43; Settler 2011, 90).

Daily Scrum -tapaamiset edistävät kommunikointia, auttavat tunnistamaan ja poistamaan esteitä, kannustavat nopeaan päätöksentekoon sekä parantavat näkyvyyttä (Cho ym. 2008, 191.) Daily Scrum -tapaamisessa ei keskustella suunnittelusta tai selvittää ongelmia, koska aika ei riitä. Jokaisen tiimin jäsenen täytyy saapua ajoissa paikalle, sillä tapaaminen alkaa täsmälleen sovittuun aikaan riippumatta siitä, kuka on paikalla ja kuka ei (Schwaber & Beedle 2002, 43 - 44).

Scrum Master on vastuussa onnistuneen Daily Scrumin vetämisestä. Hän pitää Daily Scrumin lyhyenä pakottamalla kaikki noudattamaan sääntöjä ja varmistamalla, että jokainen puhuu lyhyesti. Tämä edellyttää melkoista määrää rohkeutta, koska säännöt koskevat kaikkia (Schwaber & Beedle 2002, 40 - 42).

Scrum Masterin korkein prioriteetti on esteiden poistaminen. Auttaakseen Scrum Masteria esteiden poistamisessa, tiimin vastuulla on kertoa mitkä esteet estävät heidän edistymistään. Tämä tapahtuu joka päivä Daily Scrumissa. Tämä systemaattinen jatkuva palaute varmistaa, että Scrum Master tietää koko ajan, mikä estää tiimiä ja voi poistaa näitä esteitä (Settler 2011, 94). Jos esteitä ei heti selvitetä, tiimi raportoi seuraavana päivänä, että este on edelleen olemassa. On erittäin huono merkki, jos tiimi ei raportoi ratkaisemattomista esteistä. Tämä tarkoittaa usein sitä, että tiimin jäsenet ovat menettäneet uskonsa siihen, että Scrum Master voi ratkaista esteen. Jos jostakin hyvästä syystä estettä ei voida poistaa, Scrum Masterin täytyy kertoa tämä seuraavan päivän Daily Scrumissa (Schwaber & Beedle 2002, 45).

Johdon kannattaisi myös osallistua Daily Scrumiin, koska kuuntelemalla tarkasti, johto voi ymmärtää, mitä tiimi on tekemässä ja kuinka todennäköistä onnistuminen on. Osallistuminen Daily Scrumiin on helpompaa ja informatiivisempaa kuin raporttien lukeminen. Koska Scrum on suora ja avoin ja Daily Scrumin raportointiväli on vain 24 tuntia, johdon on helppo jatkuvasti tarkkailla tiimiä ja sen edistymistä (Schwaber & Beedle 2002, 40).

Daily Scrum - tapaamista voi parantaa muun muassa siten, että tapaamisessa seistään, tapaaminen kestää maksimissaan 15 minuuttia, kaikki esitellään, kukaan ei kirjoita mitään sekä keskitytään kysymyksiin kaksi ja kolme (Ness Technologies 2010, 5.)

2.6.5 Sprint Review

Sprint Review - tapaaminen on kolmenkymmenen päivän välein ja sen koordinoinnista ja johtamisesta vastaa Scrum Master. Tapaamiseen kuuluu demotilaisuus ja Retrospective. Demotilaisuuden aikana tiimi esittelee johdolle, asiakkaille, käyttäjille ja Product Ownerille tuotteen Sprintin aikana tehdyt lisäykset (Schwaber & Beedle 2002, 54; Elssamadisy 2009, 103.) Demojen avulla asiakkaat näkevät tehdyn tuotoksen ja sen hyödyt ja sitä kautta pääsevät arvioimaan järjestelmää. Edistymisen avulla eli näkyvyydellä ja jatkuvuudella saavutetaan luottamusta. Demotilaisuudessa tiimi esittelee järjestelmän ja siihen mennessä rakennetun toiminnallisuuden ja asiakkaat voivat esittää mielipiteensä ratkaisujen toimivuudesta (Elssamadisy 2009, 103). Tiimi kertoo epäonnistumiset sekä onnistumiset sekä missä vaiheessa tuotteen tai järjestelmän rakentaminen on. Tämän jälkeen he pystyvät päättämään, mitä seuraavaksi tehdään. Johto tulee tapaamiseen saadakseen selville, mitä tiimi on pystynyt rakentamaan annetuilla resursseilla. Asiakkaat tulevat tapaamiseen arvioidakseen, mitä mieltä he ovat aikaansaannoksista. Product Owner tulee katsomaan, kuinka paljon toiminnallisuutta rakennettiin. Muut kehittäjät tulevat katsomaan, mitä tiimi pystyi toteuttamaan valitulla teknologialla (Elssamadisy 2009, 104). Kaikki haluavat nähdä, mitä tiimi rakensi, kuulla minkälainen Sprintti oli, miten teknologia toimi, mitä oikopolkuja jouduttiin käyttämään, mitä lisäyksiä pystyttiin tekemään ja mitä voidaan tehdä seuraavaksi (Schwaber & Beedle 2002, 55).

Tapaaminen alkaa yleensä sillä, että Scrum Master esittelee tiiviin katsauksen Sprintistä. Sprintin tavoitteita ja Product Backlogia verrataan Sprintin todellisiin tuloksiin ja eroavaisuuksien syistä keskustellaan. Tapaamisen aikana kaikki muodostavat käsityksen esitettävän tuotteen toimivuudesta omassa ympäristössään. Tämän jälkeen mietitään mitä toiminnallisuutta voidaan lisätä seuraavassa Sprintissä. Tapaamiseen ei pitäisi valmistautua erityisesti ja siksi yleensä PowerPoint- ja sen kaltaiset esitykset ovat kiellettyjä. Sprint Review -palaverit ovat hyvin informatiivisia ja keskeistä on rakennettu tuote. Kysymykset, havainnot, keskustelu ja ehdotukset ovat sallittuja ja suositeltavia. On kuitenkin hyvä muistaa, että tapaaminen on informoiva, ei arvosteleva. Kaikkien täytyy saada käsitys tuotteen lisäyksistä, koska he tarvitsevat tiedon Sprint Planning -tapaamiseen (Schwaber & Beedle 2002, 55 - 56).

Sprint Review -tapaamisen jälkeen tiimi pitää Scrum Masterin kanssa Retrospective-tapaamisen, jossa keskustellaan siitä, mikä meni hyvin, mikä ei mennyt hyvin ja mitä parannuksia voitaisiin tehdä seuraavaan Sprinttiin. Tämä on erityisen tärkeä mahdollisuus tiimille keskittyä kokonaisvaltaisesti suoritukseensa ja tunnistaa strategioita, joiden avulla se voi parantaa prosessejaan (Settler 2011, 105; Elssamadisy 2009, 109). Samoin se on arvokas tilaisuus Scrum Masterille, joka voi huomata yleisiä esteitä ja työskennellä niiden poistamiseksi (Settler 2011, 105).

Retrospective -tapaamisessa Scrum-tiimi ja Scrum Master tapaavat ja analysoivat Sprinttiä sekä miettivät miten jatkossa voitaisiin parantaa tiimin ympäristöä ja prosesseja. Retrospective -tapaamisen tarkoituksena on arvioida ihmisiä, heidän vuorovaikutustaan sekä heidän käyttämiään työkaluja kun taas demo keskittyy arvioimaan tuotetta joka rakennettiin (Elssamadisy 2009, 109).

2.7 Ketterän Scrum - menetelmän roolit

2.7.1 Product Owner

Cohnin (2010, 125 - 126) mielestä Product Owner on henkilö, joka näyttää tiimille oikean päämäärän ja varmistaa, että tiimi työskentelee tämän päämäärän saavuttamiseksi. Cohnin mukaan on vaikeaa koota perusteellista listaa Product Ownerin vastuista, vaan on helpompaa ajatella kahta asiaa, jotka Product Owner tuottaa tiimille. Nämä kaksi asiaa ovat: visio ja rajat. Visio ja rajat voidaan ajatella kilpailevina näkökulmina projektissa. Visio näyttää mitä tuotteesta voi tulla ja rajat kuvaavat ne realiteetit, miten visio täytyy toteuttaa.

Product Ownerin vastuisiin kuuluu siis tuotteen vision laatiminen ja siitä kommunikoiminen. Product Ownerin täytyy selventää visio tiimille ja sen hän tekee luomalla, ylläpitämällä ja priorisoimalla Product Backlogia (Cohn 2010, 126; Settler 2011, 103). Product Owner priorisoi Product Backlogia miettimällä mitkä toiminnot tuottavat korkeimman liiketoiminta-arvon (Settler 2011, 103.) Parhaimpia tiimejä ovat ne joiden motivaation on syyttänyt Product Ownerin kanssa jaettu ”mukaansatempaava visio. Kenelle me myymme tuotetta? Mikä on ainutlaatuista tuotteessamme? Mitä meidän kilpailijamme tekevät? Miten meidän tuotteemme kehittyä ajan kuluessa?”. Kysymykset voivat olla erilaisia riippuen tuotteesta tai palvelusta mutta jaettu visio on erittäin tärkeää tiimin motivaation kannalta (Cohn 2010, 126).

Product Owner tekee tiivistä yhteistyötä asiakkaiden ja tiimin kanssa varmistaakseen, että asiakkaiden näkemykset ja heidän halunsa heijastuvat tuotokseen ja että tuote toteutetaan ajallaan ja budjetin rajoissa. Product Ownerin täytyy siis olla rajoituksetta tiimin käytettävissä osoittaakseen suunnan ja vastataksien kysymyksiin (Settler 2011, 104.) Olemalla saatavilla

Product Owner näyttää, että hän on sitoutunut tiimiin. Tiimi ei voi pitkään odottaa vastukseen kysymyksiinsä, vaan sen täytyy saada vastaukset nopeasti, jotta sen laatima rytmi ei häiriinny. Parhaimmat Product Ownerit tekevät mitä tahansa, jotta pystytään rakentamaan paras mahdollinen tuote (Cohn 2010, 130).

Kun tiimi oppii tuntemaan Scrumin ja on hyvä siinä, sen työtahti nopeutuu, tehokkuus kasvaa ja tällöin kasvavat myös vaatimukset Product Ownerin ajalle. Näin tapahtuu todennäköisesti vaikka tiimi oppisi tuntemaan liiketoiminta-aluetta ja ottaisi enemmän vastuuta itselleen. Jokaisella tiimillä täytyisi olla myös oma Product Owner, koska Product Ownerin työ on hyvin haasteellista. Yksi osa työtä on työskentely tiimin kanssa, jotta tuote saadaan rakennettua ja toinen osa on asiakkaiden kanssa keskustelu ja markkinatrendien seuraaminen (Cohn 2010, 128 - 129).

Product Ownerien päämääränä pitää olla oikean liiketoiminta-arvon tuottaminen, ja heidän täytyy tukea tiimiä luomaan sellaisia ratkaisuja joiden avulla tätä arvoa saadaan tuotettua. Product Ownerit kuuntelevat ja arvioivat asiakkaiden tarpeita ja siksi heillä täytyy olla vahva liiketoiminta-aisti sekä ymmärrys asiakkaistaan ja markkinoistaan. Lisäksi heidän täytyy myös ymmärtää, mikä on teknisesti mahdollista (Settler 2011, 127).

Product Ownerilla täytyy olla myös hyvät kommunikaatiotaidot, koska hän on vuorovaikutuksessa käyttäjien, asiakkaiden, organisaation johdon, yhteistyökumppanien ja muiden tiimiläisten kanssa. Taidokkaat Product Ownerit pystyvät tuomaan saman informaation kaikille erisidosryhmille samalla kuitenkin räätälöimällä tiedon siten, että se sopii jokaiselle sidosryhmälle parhaalla mahdollisella tavalla. Product Ownerin täytyy myös kuunnella käyttäjiä, asiakkaita sekä varsinkin tiimiä (Cohn 2010, 130).

Product Ownerin pitää olla jonkun valtuuttama siten, että hänellä on valta tehdä päätöksiä. Jos jotkut ulkopuoliset kumoavat jatkuvasti Product Ownerin päätöksiä, tiimin jäsenet oppivat menemään näiden luokse kysymään tärkeitä kysymyksiä (Cohn 2010, 131). Jotta Product Owner voi onnistua, kaikkien organisaatiossa täytyy kunnioittaa hänen päätöksiään. Kenelläkään ei ole oikeutta käskää Scrum tiimin tai tiimien työskennellä toisessa järjestyksessä eikä Scrum tiimi saa kuunnella ketään, joka sanoo toisin. Kaikki Product Ownerin päätökset ovat näkyviä, koska ne vaikuttavat Product Backlogin priorisointeihin. Tämä edellyttää, että Product Owner tekee parhaansa ja tämän takia rooli on sekä vaativa että palkitseva (Schwaber & Beedle 2002, 34 - 35).

2.7.2 Scrum Master

Scrum Masterin ensisijainen tehtävä on poistaa kaikki mahdolliset Sprintin tavoitteiden tiellä olevat esteet. Scrum Master ei ole johtaja tai orjapiiskuri eikä koskaan kommentoi työtä tii-

min puolesta. Scrum Masterilla ei myöskään ole varsinaista määräysvaltaa Scrum-tiimiin tai Product Owneriin (Settler 2011, 100). Scrum Masterin työ ei ole myöskään rajattu vain tiimiin, vaan hänellä on vastuu auttaa Product Owneria maksimoimaan tuottavuutta. Tähän voi kuulua Product Backlogin ja versiosuunnitelman ylläpidossa auttaminen tai muiden Scrum artefaktien seuraaminen (Schwaber & Beedle 2002, 31.)

Scrum Master ottaa esteet vastuulleen ja työskentelee kaikkien organisaation jäsenten kanssa esteiden poistamiseksi. Scrum Masterilla täytyy olla johdon täysi tuki ja sitoutuminen ja ennen kaikkea valtaa esteiden poistamiseen. Jos johto on erimieltä Scrum Masterin tekemisistä, täytyy johdon tarjota ehdotuksia, ohjausta ja valmennusta ja kaikesta huolimatta tukea Scrum Masteria. Valtuuksien antaminen Scrum Masterille, jotta hän voi poistaa esteitä, voi olla organisaatiolle arka paikka. Jos organisaatio mahdollistaa valtuuksien antamisen Scrum Masterille ja arvostaa Scrum Masterin pyrkimyksiä, Scrum toimii niin kuin sen pitääkin ja tuottavuus kasvaa (Schwaber & Beedle 2002, 141 - 142).

Scrum Masterin vastuulla on varmistaa, että Scrumin prosessia noudatetaan. Hyvä Scrum Master pystyy ja on halukas omaksumaan vastuullisuutta. Se ei tarkoita, että hän olisi vastuussa projektin onnistumisesta, vaan se on yleensä ottaen jaettu tiimin kanssa. Scrum Masterin vastuulla on kuitenkin maksimoida tiimin suoritustehoa ja avustaa tiimin jäseniä omaksumaan ja käyttämään Scrumia. Hyvä Scrum Master on nöyrä ja tekee mitä tahansa mahdollista auttaakseen tiimiä saavuttamaan tavoitteet ennemmin kuin asettaa omat tarpeensa ensimmäiseksi (Cohn 2010, 118 - 119).

Hyvä Scrum Master työskentelee varmistaakseen, että tiimin kesken vallitsee yhteistyökykyinen kulttuuri. Scrum Masterin täytyy varmistaa, että tiimin jäsenet tuntevat voivansa nostaa asioita avoimeen keskusteluun sekä tuntevat saavansa tähän tukea. Scrum Masterin täytyy tuntea samanasteista sitoutumista projektiin ja sen hetkisen Sprintsin tavoitteisiin kuin tiimin (Cohn 2010, 119 - 120).

Onnistunut Scrum Master vaikuttaa sekä tiimiin että sen ulkopuolisiin. Scrum Masterin täytyy tietää, miten käyttää vaikutusvaltaansa vaikuttamatta diktaattorilta. Scrum Master, joka tietää kuka tekee organisaation päätökset, miten päätökset tehdään jne., voi olla valtti tiimille. Sen lisäksi, että Scrum Masterilla on täysi ymmärrys ja kokemus Scrumista parhailla Scrum Masterilla on myös tekninen, markkinoinnillinen tai jokin muu erikoistietämys jonka avulla Scrum Master pystyy auttamaan tiimiä saavuttamaan tavoitteen. Scrum Masterin ei kuitenkaan välttämättä tarvitse olla markkinointi- tai ohjelmointiguru, vaan hänen täytyy tietää tarpeeksi paljon molemmista alueista toimiakseen tehokkaasti tiimiä johtaessaan (Cohn 2010, 120).

2.7.3 Scrum-tiimi

Scrum-tiimin vastuulla on saavuttaa tavoitteet, joihin se sitoutui Sprint Planning - tapaamisessa. Sitoutuminen koskee koko tiimiä, ei vain yksittäistä tiimi jäsentä. Joissakin organisaatioissa voi olla vaikea muuttaa kulttuuria, jossa ajatellaan, että ”minä olen vastuussa ja sitoutunut suorittamaan omat tehtäväni” kulttuuriksi, jossa ideana on tehtävien jakaminen ja se, että koko tiimi vastaa tehtävistä. Ennen kuin tämä on muuttunut, tiimien on vaikea saada tehtyä kaikkia niitä tehtäviä, jotka Sprinttiin valittiin. Kun koko tiimi on vastuussa, tiimin jäsenen, joka on edellä aikataulusta, auttaa niitä jotka ovat jäljessä toivoen, että kaikki saavat tehtävänsä tehtyä ajoissa (Cohn 2010, 204.)

Onnistuneimmissa tiimeissä tiimin jäsenet tekevät kaiken tarvittavan tavoitteiden saavuttamiseksi. Tiimin jäsenet ovat ymmärtäneet tiimin tavoitteet ja sen, mitä onnistuminen tarkoittaa sekä tiimille että asiakkaille. He ymmärtävät, että onnistuminen tarkoittaa virheiden hyväksymistä ja niistä oppimista, ja siksi tiimin jäsenet tuntevat olonsa hyväksi jakaessaan kokemuksia (Elssamadisy 2009, 17.) Jokaisen tiimin jäsenen täytyy myös käyttää omaa erikoisosaamistaan kaikissa ongelmissa, koska tästä syntyvä synergia parantaa laatua ja tuottavuutta (Schwaber & Beedle 2002, 37).

Tiimillä saattaa aluksi olla vaikeuksia Scrumin omaksumisessa. Se joutuu kamppailemaan siitä kuinka paljon yksityiskohtia Product Backlogissa täytyy olla, kuinka paljon työtä saadaan valmiiksi Sprinteissä, miten työskennellä hyvin yhdessä Sprintin aikana jne. Tiimin jäsenet oppivat uusia käytäntöjä ja uusia tapoja työskennellessään yhdessä. Kun tiimistä tulee nopeampi, se pystyy saamaan valmiiksi enemmän töitä Sprintin aikana (Cohn 2010, 128).

Vain tiimi voi arvioida minkä määrän se voi toteuttaa seuraavan 30 päivän aikana. Tiimillä on valta tehdä kaikki tarvittavat päätökset ja se voi pyytää poistamaan kaikki mahdolliset esteet, jotka voivat estää sen edistymistä. Vaikka tiimillä on valtuus päättää miten se tekee työnsä, se vastuulla on kuitenkin käyttää ja noudattaa olemassa olevia ohjeita, standardeja, sääntöjä, arkkitehtuuria ja teknologiaa. Näin varmistetaan, että projektin tuotokset ja Scrum-tiimi sopivat organisaatioon. Noudatettavien standardien ja sääntöjen täytyy olla selvillä ennen Sprintin aloitusta (Schwaber & Beedle 2002, 38 - 39).

Tiimin ihannekoko on seitsemän henkilöä, plus tai miinus kaksi (Schwaber & Beedle 2002, 36; Settler 2011, 107.) Pienemmät tiimit voivat rajoittaa vuorovaikutuksen määrää ja heikentää tuottavuuden saavuttamista. Suuremmatkaan tiimit eivät toimi hyvin koska tuottavuus laskee ja Daily Scrumin vetäminen voi olla hankalaa. Tiimit kannattaa jakaa useammaksi tiimiksi, jos henkilöitä on yli kahdeksan. Kukin tiimi valitsee omien erikoistaitojensa mukaisesti priorisoi-

dusta Product Backlogista niitä asioita, jotka se hallitsee parhaiten (Schwaber & Beedle 2002, 37).

Avoin työympäristö antaa tiimille useita kommunikointikanavia. Henkilökohtainen kommunikatio on tehokkain kommunikaation keino ja se rakentaa luottamusta tiimin välillä ja lisää yhteenkuuluvuuden tunnetta ja helpottaa itseorganisoitumista (Schwaber & Beedle 2002, 39.) Toisistaan kaukana olevilla tiimeillä on suuri virhealttius ja lisäksi hidas kommunikaatio hidastaa kehitystä ja aiheuttaa ”Me vastaan He” - ympäristön. Elleivät tiimit voi olla samassa paikassa, täytyy yrittää luoda luottamusta tiimien kesken ja käyttää selkeitä kommunikointikanavia kuten videolaitteita (Elssamadisy 2009, 119 - 120).

Tiimien sijaitessa toisessa paikassa on hyvä noudattaa seuraavia parhaita käytäntöjä.

- Kaikkien tiimien jäsenten täytyy olla paikalla projektin käynnistyessä ja ensimmäisessä Sprintissä. Lisäksi on suositeltavaa, että tiimien jäsenet olisivat paikalla jokaisessa versionjulkaisussa tai isoissa toimituksissa.
- Tiimien täytyy saada järjestää virallisten tapaamisten lisäksi myös vapaita tapaamisia koska ne auttavat tiimejä parantamaan tiimien välisiä suhteita projektin aikana. Tämä kasvattaa projektin kustannuksia mutta kokemus on osoittanut, että nämä tapaamiset parantavat tehokkuutta, sovelluksen laatua ja saavat aikaan lyhyempiä syklejä.
- Tiimien väliseen kommunikointiin täytyy käyttää työkaluja kuten videolaitteita, puhelimia ja erilaisia ohjelmistoja jotta tiimit voivat jatkuvasti olla vuorovaikutuksessa ja kommunikoida toistensa kanssa (Ness Technologies 2010, 5).

Tiimiä koottaessa täytyisi varmistaa, että siihen valitaan henkilöitä, jotka omaavat erilaisia taitoja. Tiimissä on hyvä olla esimerkiksi kokeneempia ja vähemmän kokeneempia ohjelmoijia, jolloin vähemmän kokeneet voivat oppia kokeneemmilta. Tiimissä on myös hyvä olla sellaisia henkilöitä jotka tuntevat hyvin liiketoiminta-alueen että sellaisia jotka eivät tunne. Näin saadaan ainakin vähitellen levitettyä liiketoiminta-alueen tietämystä organisaatiossa (Cohn 2010, 190).

Scrum-tiimi voi olla tehokkaampi vastaamaan muutoksiin jos se pystyy vähentämään informaation kustannuksia sekä vähentämään aikaa päätöksen teon tekemisen ja päätöksen teon tekemisen seurauksen näkemisen välillä. Informaation kustannuksia tiimi voi vähentää työskentelemällä yhteisessä tilassa, kommunikoimalla henkilökohtaisesti sekä tiimin sopuisuutta parantamalla. Aikaa päätöksestä palautteeseen tiimi voi vähentää ottamalla käyttäjäasiantuntijat mukaan projektiin tai vielä parempi mukaan tiimiin sekä työskentelemällä inkrementaalaisesti. Käyttäjäasiantuntijoiden mukana olo auttaa kehittäjiä saamaan nopeaa palautetta ja käyttäjäasiantuntijat näkevät kasvavan tuotteen aikaisessa vaiheessa, oppivat ymmärtä-

mään kehittäjien väärinkäsityksiä sekä näkevät mitkä heidän vaatimuksista eivät toimikaan niin hyvin käytännössä kuin oli ajateltu (Cockburn & Highsmith 2001, 131).

3 Tutkimusmenetelmän kuvaus

3.1 Tapaustutkimus (case study)

Tapaustutkimuksessa tutkitaan yhtä tapausta tai useampia tapauksia. Tutkittava tapaus (case) voi olla esimerkiksi yksilö, ryhmä tai yritys, jota tarkastellaan sen omassa kontekstissaan eli ympäristössään (Kananen 2008, 84; Yin 2009, 4; Benbasat, Goldstein & Mead 1987, 370). Tapaustutkimus voi olla kuvailevaa, teoriaa testaavaa tai teoriaa luovaa (Järvinen & Järvinen 2004, 75; Darke, Shanks & Broadbent.) Tapaustutkimuksessa voi käydä myös niin, että mitään uutta teoriaa ei synny, mutta tapauksesta tai tapauksista sekä niiden ympäristöstä voidaan saada parempaa ymmärrystä (Järvinen & Järvinen 2004, 75.)

Tapaustutkimuksessa on Kananen (2008, 84) mukaan kaksi edellytystä. Ensimmäinen edellytys on ilmiön tarkasteleminen monien eri tietolähteiden avulla. Toinen edellytys on tutkimusaineiston esittäminen niin, että muut pystyvät tarkistamaan todisteketjun (Kananen 2008, 128). Lukijan pitää pystyä seuramaan tutkimuskysymyksiä johtopäätöksiin ja päinvastoin ja esitetävien todisteiden pitää olla samat jotka kerättiin tiedon keräysvaiheessa (Dubé & Pare 2003, 618.) Myös tiedon asiayhteys ja monipuolisuus täytyy esittää ja todisteketjun pitää olla aukoton (Benbasat ym. 1987, 374.)

Tapaustutkimuksen tutkimusaineistona toimivat dokumentit, arkistot, haastattelut ja havainnoinnit (Järvinen & Järvinen 2004, 75; Yin 2009, 11.) Olennaista on käyttää ja yhdistää monia eri tietolähteitä. Tätä kutsutaan triangulaatioksi (Yin 2009, 11; Kananen 2008, 84). Triangulaation avulla kasvatetaan tutkimus tulosten sisäistä validiteettia. Tärkeää on myös kuvata, kuinka triangulaatio saavutetaan (Dubé & Pare 2003, 627.)

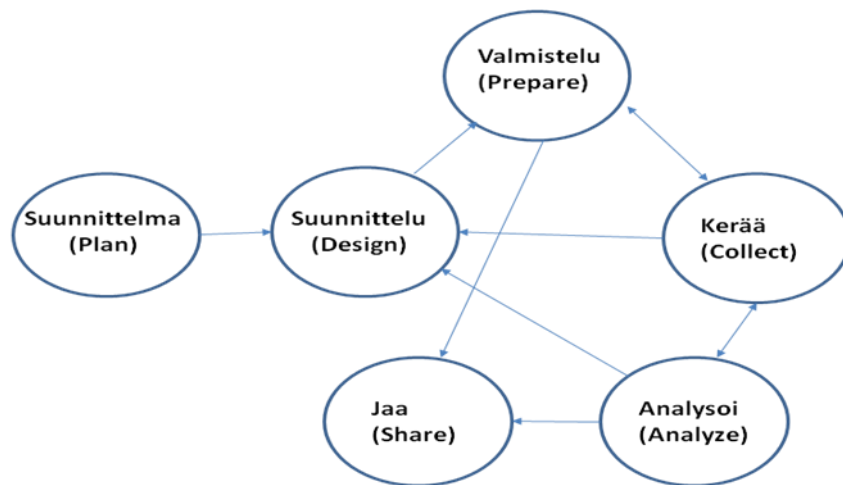
Tapaustutkimuksen ”lähtökohdat ovat tieteellisen tutkimuksen traditiossa”. Tapaustutkimus soveltuu hyvin lähestymistavaksi myös silloin, kun tarkoituksena on tuottaa kehittämisohdotuksia ja -ideoita (Ojasalo, Moilanen & Ritalahti 2009, 52).

Tapaustutkimus sopii kolmesta syystä hyvin myös tietojärjestelmäalan tutkimukseksi. Ensimmäiseksi siksi, että tutkija voi tutkia tietojärjestelmiä niiden omassa ympäristössä, oppia tekniikan nykytilasta ja luoda teoriaa käytännöistä. Toiseksi siksi, että tapaustutkimuksen avulla tutkija voi vastata ”miksi” ja ”miten” kysymyksiin ymmärtääkseen tutkittavan prosessin ympäristön ja monimutkaisuuden ja kolmanneksi siksi, että tapauksen tai tapauksien avulla voi-

daan tutkia aluetta mistä on tehty vain muutamia aikaisempia tutkimuksia (Benbasat ym. 1987, 370).

3.2 Tutkimuksen toteutus

Tutkimus toteutettiin käyttämällä Yinin (2009) tapaustutkimuksen prosessia joka sisältää seuraavassa kuvassa esiintyvät kuusi vaihetta.



Kuva 3. Tapaustutkimuksen vaiheet Yinin (2009) mukaan

Seuraavissa luvuissa esitellään nämä kuusi vaihetta sekä tutkimuksen eteneminen näiden vaiheiden mukaisesti.

3.2.1 Tutkimussuunnitelma

Tutkimuksen suunnitelmavaiheeseen kuuluu tutkimusmenetelmien valinta, tutkimusmenetelmien vertailu ja tutkimuskysymysten tunnistaminen (Yin 2009, 2; Kananen 2008, 61.) Yin (2009, 2) lisää vaiheeseen vielä perustelut tapaustutkimuksen tekemiselle ja tapaustutkimuksen vahvuuksien ja heikkouksien ymmärtämisen ja Kananen (2008, 61) lisää tutkimusongelman ja tutkimuksen tavoitteiden määrittämisen.

Tutkimusmenetelmäksi valittiin tapaustutkimus Yinin (2009) mukaan koska, tutkimuksessa on tarkoitus tutkia ketterään Scrum - menetelmään liittyviä asioita kolmen tapauksen avulla. Tapauksien avulla saatuja näkemyksiä pyritään tukemaan teoreettisten väittämien avulla. Näin saavutetaan usean eri tietolähteen käyttö mikä on Yinin (2009, 11) mukaan yksi tapaustutkimuksen voima. Tapaukset ja niiden toimintaympäristö kuvataan seuraavassa luvussa.

Tutkimuskysymyksiä tunnistamisessa on tärkeää varmistaa, että tutkimuskysymykset ovat tärkeitä ja arvokkaita sekä tutkimukselle että organisaatiolle. Kysymysten täytyy olla mielenkiintoisia ja tärkeitä sekä sellaisia, että niihin pystytään saamaan tarpeelliset vastaukset (Darke ym. 1988, 280 - 281). Tutkimuskysymyksiä syntyi kaksi eli miten Scrumin pilottikäyttö onnistui Samlinkin sähköisen työpöydän pilottiprojektissa ja mitä täytyisi jatkossa kehittää ja parantaa jotta Scrumin käyttö ja sen käyttöönotto olisi tulevissa Samlinkin sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa.

Tutkimusmenetelmän valinnan ja tutkimuskysymyksiä tunnistamisen jälkeen tehtiin opinnäytetyön tutkimussuunnitelma, jossa kerrottiin tutkimusongelmasta, teoreettisesta taustasta, tutkimusmenetelmästä, odotetuista tuloksista ja niiden hyödyistä, aikataulusta sekä kuvattiin mahdolliset ongelmat.

3.2.2 Tapauksien ja toimintaympäristön kuvaus

Samlink on perustettu vuonna 1994 ja se on erikoistunut finanssialan tietotekniikan palveluihin. Samlink tarjoaa asiakkailleen korkealaatuisia IT-järjestelmiä ja palveluita ja sillä on vahva kokemus pankkitoiminnan teknisistä ratkaisuista ja palveluprosesseista (Samlinkin vuosikertomus 2010.) Samlinkin asiakkaita ovat pankit ja sijoituspalveluja tarjoavat yritykset. Samlinkin asiakaspalvelumalli ja kaikille asiakkaille yhteiset järjestelmät mahdollistavat kilpailukykyiset hinnat. Asiakkaiden kilpailuetu syntyy perusratkaisujen muuntelulla sekä asiakaskohtaisesti räätälöityjen tuotteiden ja palvelujen tuottamisella. Samlinkin asiakkaille on tärkeää järjestelmien kustannustehokkuus ja tähän vastataan jatkuvalla toimintaprosessien kehittämällä (Samlinkin kotisivut 2011).

Samlink kehittää ja ylläpitää erilaisia sovelluksia ensisijaisesti pankkien toimihenkilöiden käyttöön. Sovelluksien avulla pankkien toimihenkilöt suoriutuvat päivittäisistä työtehtävistään. Sovelluskehityksen apuna on käytetty systeemyömenetelmää, joka perustuu vesiputousmalliin. Menetelmän perusmalliin kuuluu kuusi vaihetta: vaatimusmäärittely, määrittely, suunnittelu, toteutus, testaus ja käyttöönotot.

Tapauksien toimintaympäristö on viime vuoden marraskuussa käynnistynyt sähköisen työpöydän pilottiprojekti, missä rakennetaan pankkien toimihenkilöiden käyttöön sähköistä työpöytää. Sähköinen työpöytä kuuluu Samlinkin NEXT-hankkeeseen. Hankkeen tarkoituksena on mahdollistaa kaikkien pankkien operatiivisten järjestelmien käyttäminen yhden käyttöliittymän kautta. Kehitystyötä tehdään yhdessä asiakkaiden kanssa ja sen perimmäisenä ideana on prosessien sujuvuuden lisääminen ja pankkien toimihenkilöiden päivittäisen työn helpottaminen (Samlinkin vuosikertomus 2010, 9 - 10.)

Sähköisen työpöydän pilottiprojektissa päätettiin kokeilla ketterää Scrum -menetelmää (Simola 2011.) Projektissa oli mukana Samlinkin asiakkaita sekä muita asiantuntijoita. Tästä syntyi ajatus opinnäytetyön tekemiseen, koska ketterä Scrum -menetelmä kiinnosti aiheena ja tutkimuksessa voitiin hyödyntää projektiin osallistuneiden henkilöiden näkemyksiä. Näitä näkemyksiä haettiin haastattelujen ja sähköisen kyselylomakkeen avulla. Tämän lisäksi haluttiin selvittää myös yrityksen näkemys ketterästä Scrum -menetelmästä ja siksi haastateltiin myös Sovelluspalvelut -yksikön johtajaa. Edellisten johdosta tutkimuksen tapauksiksi muodostuivat siis yritys sekä projektissa mukana olleet asiakkaat ja muut asiantuntijat. Näiden tapauksien näkemysten avulla pyritään saamaan vastaukset tutkimusongelmaan ja tutkimuskysymyksiin ja näkemyksiä pyritään tukemaan teoreettisten väittämien avulla.

3.2.3 Tutkimuksen suunnittelu

Tutkimuksen suunnitteluun kuuluu tunnistaa tutkimuksen unit of analysis. Unit of analysis on sama kuin tapauksen määrittelemineen, eli se tarkoittaa sen ilmiön tunnistamista, jota tutkimuksessa aiotaan tutkia. Tämä ilmiö voi olla esimerkiksi yksilöt, ryhmät tai yritys (Yin 2009, 24). Unit of analysisia mietittäessä täytyy miettiä mihin tutkimus keskittyy ja tutkia tutkimuskysymyksiä. Nämä usein ilmaisevat mikä on tutkimuksen unit of analysis (Benbasat ym. 1987, 372.) Tarkoituksen mukaisen Unit of analysisin täytyy myös mahdollistaa tutkimuskysymyksiin vastaaminen riittävän tarkalla tasolla (Darke ym. 1998, 280.)

Unit of analysis määrittelyn lisäksi suunnitteluvaiheeseen kuuluu teorian kehittäminen, tapaustutkimuksen tekemisen suunnittelu (yksi tapaus, monta tapausta, holistinen, sulautettu) sekä niiden menettelytapojen määrittelemineen, joiden avulla ylläpidetään tapaustutkimuksen laatua. Lisäksi kannattaa jo suunnitteluvaiheessa selvittää, minkälaisia analyysimenetelmiä aikoo käyttää (Yin 2009, 23.)

Suunnitteluvaihe käynnistyi tutkimuskysymysten pohjalta ja tätä kautta miettimällä mikä tai mitkä ovat tämän tutkimuksen tapaus tai tapaukset ja mikä on ”unit of analysis”. Koska tutkimuksessa selvitetään ilmiötä eri näkemyksien avulla, tutkimuksen tapauksiksi nousivat nämä näkemykset eli yritys sekä projektissa mukana olleet asiakkaat ja asiantuntijat. Monen tapauksen tutkimus sopii hyvin silloin kun tutkimuksen on tarkoitus olla kuvaileva, teoriaa rakentava tai teoriaa testaava (Benbasat ym. 1987, 373.) Tämä tutkimus on kuvaileva, koska sen tarkoituksena on kuvata tapauksien haastatteluista ja kyselyistä esiin nousseita näkemyksiä sekä tukea näitä näkemyksiä teoreettisten väittämien avulla. Tapaukset ja niiden toimintaympäristö on kuvattu luvussa tapauksien ja toimintaympäristön kuvaus. Tutkimuskysymyksiin haetaan vastauksia näiden tapauksien vastauksien perusteella, ja vastauksia pyritään tukemaan teoreettisten väittämien avulla. Tämän takia tutkimuksen unit of analysis on nämä

eri näkökulmat eli kirjallisuus, yritys sekä projektissa mukana olleet asiakkaat ja muut asiantuntijat.

”Unit of analysis” -määrittelyn jälkeen selvitettiin erilaisia haastattelumenetelmiä ja niistä päädyttiin valitsemaan puolistrukturoitu haastattelu Sovelluspalvelut -yksikön johtajalle ja projektissa mukana oleville asiantuntijoille sekä puolistrukturoitu sähköinen kyselylomake asiakkaille. Puolistrukturoitu tarkoittaa sitä, että kysymykset on laadittu ennakkoon, mutta niiden järjestystä ja sanamuotoa voidaan vaihtaa haastattelun kulun mukaisesti. Haastattelun aikana voidaan jättää esittämättä joitakin kysymyksiä tai vastaavasti voidaan kysyä kysymyksiä joita tulee mieleen haastattelun kuluessa (Ojasalo ym. 2009, 97).

Haastattelu on tiedonhankinta menetelmä, missä tutkija keskustelee haastateltavan kanssa ja pyrkii omalla toiminnallaan edistämään keskustelua. Haastattelussa on hyvää se, että haastattelun aikana voi tulla esiin uusia asioita ja näitä voidaan heti pyrkiä myös tarkentamaan (Järvinen & Järvinen 2004, 146.) Haastatteluiden avulla pyritään ratkaisemaan varsinaista tutkimusongelmaa ja siksi on tärkeää, että kysymykset liittyvät tutkimuskysymyksiin (Kananen 2008, 73.)

Haastattelujen tarkoituksena oli selvittää yrityksen näkemys siihen, mitä se odottaa ketterältä Scrum -menetelmältä ja miten se aikoo vastata asiakkaiden tarpeisiin ja vaatimuksiin Scrumia käyttämällä. Projektissa mukana olevilta asiakkailta muilta asiantuntijoilta haluttiin mielipiteitä siihen, miten he ovat kokeneet ketterän Scrum -menetelmän käytön sähköisen työpöydän pilottiprojektissa, mitä kehittämistä tai parannusehdotuksia he näkevät kokeilun perusteella ja lisäksi haluttiin selvittää myös asiakkaiden odotukset ketterältä Scrum -menetelmältä. Kaikilta haluttiin myös näkemyksiä vesiputousmalliin perustuvan menetelmän ja ketterän Scrumin vertailuun sekä siihen, mitä hyvää tai huonoa vanhassa menetelmässä on.

Sovelluspalvelut -yksikön johtajan ja projektissa mukana olleiden asiantuntijoiden haastatteluissa kysymykset oli laadittu ennalta, mutta niiden järjestys ja sanamuodot vaihtelivat jonkin verran haastattelujen kulkujen mukaan. Asiakkaiden puolistrukturoidussa kyselylomakkeessa kysymykset oli muotoiltu valmiiksi, niiden järjestys oli määrätty ennalta ja ainoastaan vastauksen sisältö oli jätetty avoimeksi (Kananen 2008, 73.)

3.2.4 Tutkimuksen valmistelu

Tutkimuksen valmisteluun kuuluu tapaustutkimustaitojen hiominen, tietyn tapaustutkimuksen valmistelu ja harjoittelu, tapaustutkimuskäytäntöjen kehittäminen, mahdollisten tapausten seulominen sekä pilotti-tapaustutkimuksen tekeminen (Yin 2009, 66).

Valmisteluvaihe käynnistyi tapaustutkimusmetodiin tutustumisella eli selvitettiin mitä tapaustutkimus on ja miten tutkimus tapaustutkimuksen mukaan tehdään. Tämän jälkeen keskityttiin teoriaan ja etsittiin teoreettisia näkemyksiä Scrum-menetelmästä ja sen käytöstä sovelluskehitysprojekteissa, koska näitä näkemyksiä tarvittiin huhtikuussa suoritettuun pilotti-tapaustutkimukseen. Lisäksi teoriaan täytyi tutustua myös siksi, että saatiin ymmärrystä siihen, minkälaisia asioita haastatteluissa täytyy kysyä. Pilotti-tapaustutkimus kattoi siis teoriaosuudessa jo osittain Scrumin ideologian ja sen käyttämisen sovelluskehityksessä sekä yhden tapauksen eli yrityksen näkemykset.

Pilotti-tapaustutkimuksen avulla saavutettiin ymmärrystä ketterästä Scrum - menetelmästä ja sen käyttämisestä sovelluskehityksessä, aineiston keräämisestä sekä tuloksien määrittämisestä. Pilotti-tapaustutkimuksen avulla sai myös käsitystä siitä, minkälaisia laadullisia näkökulmia, kuten Yinin (2009, 98) mainitsevat usean lähteen käyttö ja todisteketjun säilyttäminen pitää ottaa huomioon ja miten tutkimus tulee tehdä, jotta nämä laadulliset näkökulmat sisältyvät tutkimukseen.

3.2.5 Aineiston kerääminen

Aineiston keräysvaihe sisältää tapaustutkimus käytännön seuraamisen, useiden lähteiden käyttämisen, tapaustutkimus tietokannan luomisen ja todisteketjun säilyttämisen. Näiden periaatteiden sisällyttäminen tapaustutkimukseen kasvattaa olennaisesti sen laatua (Yin 2009, 98). Tietokannan luomisella tarkoitetaan sitä, että dokumenttien pitäisi olla helposti saatavilla myöhempää tutkimusta varten. Todisteketjun säilyttäminen tarkoittaa sitä, että tutkimuksen lukijan pitää pystyä seuraamaan minkä tahansa todisteen alkuperää alkuperäisistä tutkimuskysymyksistä lopullisiin johtopäätöksiin (Yin 2009, 120 - 122).

Aineiston keräysvaiheessa määritellään, mitä tietoa tarvitaan, miten se hankitaan ja mistä se saadaan (Kananen 2008, 56.) Aineiston keräyksen tavoitteena on kerätä aineistoa tarpeeksi paljon, jotta tutkimuskysymyksiin saadaan vastaukset. Tarvittava aineisto riippuu tutkimuskysymyksistä ja unit of analysisista (Benbasat ym. 1987, 374).

Kirjallisen aineiston kerääminen tarkoittaa tiedon hankkimista kirjallisista lähteistä. Dokumentteihin kuuluvat muun muassa kirjeet, muistiot, esitykset, tutkimukset ja raportit. Arkistoihin taas kuuluvat muun muassa asiakastilastot, kuvat, päiväkirjat ja kalenterit (Järvinen & Järvinen 2004, 136). Keräysvaiheessa keskityttiin ensin pilotti-tapaustutkimuksen kirjalliseen aineistoon ja selvitettiin Internetin avulla, mitä kaikkea aineistoa löytyy ja mitä kirjoja ja artikkeleita suositellaan. Näin valittiin ne kirjat ja artikkelit, joiden avulla aineistoa lähdettäisiin keräämään. Lisäksi kerättiin tietoa yrityksestä yrityksen omilta kotisivuilta ja vuosikertomuksesta.

Teorian näkökulmasta täytyy olemassa olevasta aineisto löytää se, mitä tarvitsee oman tutkimuksen ja tehtävän kannalta. Aiheeseen pitää yleensä ensin perehtyä, jotta tutkija tietää mitä hänen täytyy kysyä tai tutkia. Tämän jälkeen voidaan kehittää tarkempia ja täsmentäviä kysymyksiä ja sitä kautta selvittää mitä teoriaa vielä tarvitaan (Ojasalo ym. 2009, 54). Pilottitapaustutkimuksen jälkeen kerättiin lisää aineistoa, jotta saatiin selville minkälaisia asioita projektissa mukana olevilta asiakkailta ja muilta asiantuntijoilta pitää kysyä. Näiden vastauksen jälkeen tiedettiin mitä teoriaa pitää vielä kerätä, jotta teorian avulla pystytään tukemaan vastaajien näkemyksiä.

Haastatteluaineistot kerättiin puolistrukturoidulla haastatteluilla sekä sähköisellä kyselylomakkeella. Sovelluspalvelut -yksikön johtajalle pidettiin haastattelu 11.3.2011, ja projektissa mukana olleille muille asiantuntijoille 11.4 - 19.4.2011 välisenä aikana. Projektissa mukana olevia muita asiantuntijoita haastateltiin yhteensä yhdeksän henkilöä, joista viisi oli sovelluskehittäjiä, yksi käyttöliittymäasiantuntija, yksi testaja, yksi Product Owner ja yksi Scrum Master. Haastattelut olivat avoimia, ja ne kestivät noin tunnin. Projektissa mukana olevien asiantuntijoiden haastattelua varten tehtiin runko aihealueista ja aloituskysymyksistä joita haastatteluissa haluttiin selvittää. Nämä aihealueet ja aloituskysymykset löytyvät liitteestä 1.

Projektissa mukana olleille viidelle asiakkaalle lähetettiin sähköinen kyselylomake 6.4.2011 ja palautukset pyydettiin 14.4.2011 mennessä. Kyselylomake sisälsi kahdeksan avointa kysymystä. Kolme ensimmäistä kysymystä liittyivät vesiputousmalliin perustuvaan sovelluskehitysmenetelmään, koska haluttiin selvittää, oliko tällainen menetelmä heille tuttu ja jos oli, mitä hyviä ja huonoja puolia he sen käytössä näkivät. Loput viisi kysymystä liittyivät ketterään Scrum -menetelmään. Asiakkaat olivat osallistuneet sähköisen työpöydän pilottiprojektissa Scrumin mukaisesti Sprint Review -tapaamisten demo-osuuksiin sekä asiakas Product Owner -palaveriin ja haluttiin selvittää, miten asiakkaat olivat nämä tilaisuudet kokeneet. Lopuksi heiltä kysyttiin, mitä he odottavat ketterän Scrum -sovelluskehitysmenetelmän käytöltä.

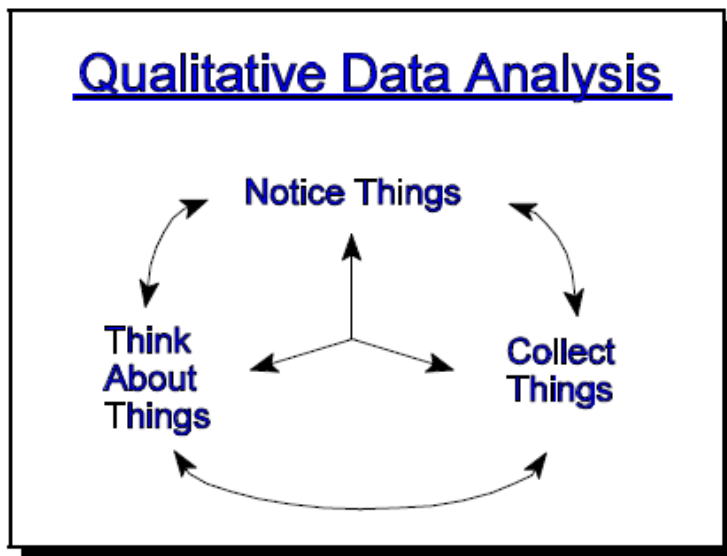
Tapaustutkimuksen tietokanta sisältää yleensä raaka materiaalin kuten haastattelujen jäljennökset, tutkijan muistiinpanot ja kerätyt dokumentit sekä koodatun tiedon kuten muistiot ja muut analysoidut materiaalit (Dubé & Pare 2003, 616.) Tietokannaksi luotiin oma kansio tietokoneelle jonne tallennettiin haastattelun runko aihealueista ja aloituskysymyksistä, kyselylomake ja niiden vastaukset sekä auki kirjoitetut haastattelut haastateltavan nimen ja haastattelupäivämäärän mukaisesti, haastattelujen yhteenvedot sekä kerätyt kirjallisuuslähteet. Lisäksi samat tiedostot tallennettiin cd-levylle.

3.2.6 Aineiston analysointi

Aineiston analysointivaihe sisältää luottamisen teoreettisiin väittämiin ja muihin strategioihin, analyysitekniikoiden harkitsemisen, vastaväitteiden selvittelyn sekä tiedon esittämisen erillään haastatteluista (Yin 2009, 126.)

Analyysistrategian valinnalla ei ole niin väliä kunhan analyysi osoittaa tärkeät ja oikeat asiat. Analyysin täytyy osoittaa, että se sisältää kaikki todisteet, sen täytyy perinpohjaisesti kattaa tutkimuskysymykset, sen täytyy näyttää, että on käytetty niin paljon ”todisteita” kuin oli saatavilla, sen täytyy osoittaa - jos mahdollista - kaikki merkittävimmät vastatulkinnot ja lisäksi analyysin pitäisi osoittaa merkittävimmät tapaustutkimuksen näkökulmat (Yin 2009, 160). Selkeä analyysistrategian kuvaus auttaa lukijaa ymmärtämään tulokset paremmin sekä osoittaa ovatko tulokset syntyneet järjestelmällisen ja täsmällisen prosessin avulla (Dubé & Pare 2003, 616.)

Aineiston analysointistrategiamallina käytettiin aineiston analyysi prosessia, joka sisältää kolme osaa: kiinnostavien asioiden huomaamisen, keräämisen ja ajattelemisen. Seuraavassa kuvassa esitetään tämä prosessi (Seidel 1998, 1).



Kuva 4. Tiedon analysointi prosessi (Seidel 1998, 2.)

Prosessilla on seuraavat ominaisuudet: iteratiivinen ja asteittainen, toistuva sekä holografinen. Iteratiivinen ja asteittainen tarkoittaa sitä, että prosessi on toistuva sykli. Esimerkiksi kun ajattelee joitakin tietoja alkaa samalla huomaamaan uusia tietoja, ja sen jälkeen alkaa kerätä ja ajatella näitä uusia tietoja. Toistuva tarkoittaa sitä, että yhdestä osasta saatetaan joutua palaamaan edelliseen osaan. Esimerkiksi tietoa kerätessä saattaa huomata uusia tietoja joita tulisi kerätä. Holografinen tarkoittaa sitä, että jokainen prosessin askel sisältää koko prosessin. Esimerkiksi kun alkaa huomata asioita suorittaa jo mielessään näiden asioiden miettimistä ja keräämistä (Seidel 1998, 2).

Ensimmäinen osio eli asioiden huomaaminen tarkoittaa havainnointia, muistiinpanojen tekemistä, dokumenttien keräämistä, haastatteluista jne. Havaituista asioista tehdään tallenne josta etsitään mielenkiintoisia asioita. Tämä tapahtuu lukemalla aineistoa moneen kertaan. Kun mielenkiintoisia asioita huomataan, ne nimetään tai ”koodataan” (Seidel 1998, 3).

Ensimmäisessä osiossa eli tiedon havainnointiosiossa käytiin läpi teoriaa eli luettiin sitä ja tehtiin pohdintaa ja analysointia siitä, mitä ja minkälaista aineistoa on olemassa ja minkälaisista aineistosta tarvitaan pilotti-tapaustutkimusta sekä opinnäytetyön tutkimusta varten. Lisäksi tarvittiin ymmärrystä siihen, minkälaisia kysymyksiä haastatteluissa ja kyselylomakkeessa täytyy esittää. Tämän tutkimisen jälkeen valittiin ne aineistot, joita analysoitaisiin ja joista yritettäisiin löytää parhaat mahdolliset teoreettiset väittämät tukemaan haastatteluista saatuja näkemyksiä ja lopulta tutkimusongelmaa ja tutkimuskysymyksiä. Aineistot myös nimettiin tiettyjen teoria-alueiden mukaan ja mielenkiintoiset asiat alleviivattiin, jotta jatkotyöskentely olisi mahdollisimman helppoa.

Haastatteluista kirjoitettiin auki ja tallennettiin ensimmäisenä Sovelluspalvelut -yksikön johtajan haastattelu, koska se oli osa pilotti-tapaustutkimusta. Tämän jälkeen se lähetettiin tarkistettavaksi yksikön johtajalle. Seuraavaksi yhdistettiin asiakkaiden vastaukset yksikön johtajan tuloksien kanssa samaan dokumenttiin. Viimeisenä kirjoitettiin auki ja tallennettiin asiantuntijoiden haastattelut käyttämällä yleiskielisiä lauseita, koska haastatteluista kiinnosti vain esiin tulleet asiat, eli sanoilla ja sanavalinnoilla ei ollut merkitystä (Kananen 2008, 80; Ruusuvoori, Nikander & Hyvärinen 2010, 427.) Tallennuksen jälkeen haastattelut lähetettiin tarkistettavaksi haastatelluille ja heillä oli myös mahdollisuus tehdä lisäkommentteja tai tarkennuksia. Lopulta yhdistettiin asiantuntijoiden haastattelujen vastaukset yhteen dokumenttiin ja merkittiin usein esiintyviä asioita eri väreillä, jotta lajittelu ja vertaaminen olisi mahdollisimman helppoa.

Toinen osio eli asioiden kerääminen tarkoittaa asioiden keräämistä ja lajittelua. Seidel (1998, 3) käyttää artikkelissaan esimerkkinä palapelin tekemistä koska siinä palapelin tekeminen yleensä aloitetaan lajittelemalla palapelin palaset tiettyyn ryhmään joka auttaa palapelin tekemistä valmiiksi. Asioiden lajittelu on siis sama asia kuin kerääminen.

Toisessa osiossa eli tiedon keräämisosiossa kirjallinen aineisto lajiteltiin sen mukaan mihin teoria-alueeseen lähde liittyi. Yrityksen ja asiakkaiden vastaukset lajiteltiin kysytyjen aihealueiden mukaan. Asiantuntijoiden vastaukset lajiteltiin alkuperäisen aihealueen rungon mukaisesti siten, että jokaisen vastaajan vastaukset tulivat dokumentissa oikean alueen kohdalle (Seidel 1998, 2).

Kolmas osio eli asioiden ajattelemisen tarkoittaa kerättyjen asioiden tutkimista. Tavoitteena on saada jokaisesta kerätyistä asiasta jonkinlainen ymmärrys, havaita sekä kerätyistä asioista että niiden ristikkäisyyksistä kaavoja ja yhteyksiä sekä tehdä yleisiä havaintoja ilmiöstä jota tutkitaan. Havaittuja asioita vertaillaan, jotta löydetään samankaltaisuuksia ja eroavaisuuksia, rakennetaan typografiaa tai etsitään sarjoja ja kaavoja (Seidel 1998, 5).

Kolmannessa osiossa eli tiedon ajattelemisiosiossa tutkittiin vastauksia. Asiakkaiden ja yrityksen vastauksista etsittiin yhtäläisyyksiä ja eroja ja analysoitiin vastauksia toisiinsa sekä kuvattiin näiden mukaan tulokset. Asiantuntija vastauksista analysoitiin jokaisen lajittelun kohdalla eniten esiin tulleita asioita ja kuvattiin tulokset näiden perusteella. Tämän jälkeen verrattiin kaikkien vastaajien tuloksia siltä osin kuin se oli mahdollista ja kerrottiin vertailut tuloksissa. Lopuksi vastauksia käytiin läpi ja analysoitiin siitä näkökulmasta, saadaanko niiden avulla vastaukset tutkimusongelmaan ja tutkimuskysymyksiin ja lisäksi samalla selvitettiin mitä teoriaa piti vielä kerätä, jotta tuloksia voidaan tukea myös teoreettisten näkemysten avulla (Seidel 1998, 5).

3.2.7 Tuloksien jakaminen

Tuloksien jakamisvaiheessa määritellään, ketkä ovat tutkimuksen vastaanottajia eli yleisöä, kirjoitetaan teksti- ja visuaalinen osio, esitetään tarpeeksi todisteita, jotta lukija voi muodostaa omia mielipiteitä sekä uudelleen luetaan ja kirjoitetaan, kunnes tutkimus on hyvin tehty (Yin 2009, 164.)

Tutkimuksen kirjoittamisessa on tärkeää miettiä ketkä ovat tutkimuksen lukijoita. Koska tämä tutkimus kohdistuu yritykseen, tärkeää on reaali maailman ja toimenpiteiden kuvaukset (Järvinen & Järvinen 2004, 175.) Koska kyseessä on opinnäytetyö, kohdeyleisönä ovat ensisijaisesti opettajat sekä muut oppilaat. Lisäksi kohdeyleisönä on myös Oy Samlink Ab, koska työ tehtiin sinne. Opinnäytetyö myös julkaistaan nykyään sähköisessä muodossa Internetissä jolloin yleisönä voi olla myös kuka tahansa aiheesta kiinnostunut.

Tuloksien jakamisvaihe alkoi jo pilotti-tapaustutkimuksessa, eli silloin kirjoitettiin ja esitettiin osa teoriaosuudesta, kuvaus yrityksestä eli Samlinkista sekä Sovelluspalvelut -yksikön johtajan haastattelun litteroidut ja analysoidut tulokset.

Tapaustutkimusraportin täytyy olla valmis ja sen täytyy sisältää tarpeeksi todisteita tuloksien tukemiseksi (Darke ym. 1998, 287.) Pilotti-tapaustutkimuksen jälkeen kirjoitettiin opinnäytetyön puuttuvat osiot eli lisättiin teoriaosuuteen loput tarpeelliset lähteet ja niiden viittaukset, kuvattiin ja kirjoitettiin tutkimusmenetelmästä laajemmin, kuvattiin ja kirjoitettiin sähköisen työpöydän pilottiprojektista, kuvattiin ja kirjoitettiin tulokset sekä kuvattiin ja kirjoitettiin

tettiin yhteenveto. Lisäksi tarkennettiin johdantoon liittyviä lukuja. Lopuksi opinnäytetyötä luettiin läpi ja tarkasteltiin kunnes se oli siinä tilassa, että se voitiin jakaa muiden kanssa.

4 Tulokset

Tässä osiossa esitetään tämän tutkimuksen tulokset. Tulokset perustuvat tietoon, jotka kerättiin haastattelujen ja kyselylomakkeen avulla 11.3-19.4.2011 Samlinkin Sovelluspalvelut-yksikön johtajalta sekä sähköisen työpöydän pilottiprojektissa mukana olleilta asiakkailta ja muilta asiantuntijoilta. Tuloksia vertaillaan saatujen vastauksien perusteella. Tarkemmat tiedot haastatteluista, kyselylomakkeesta ja niiden sisällöstä on esitetty luvussa 6.3 Tutkimuksen toteutus.

4.1 Yrityksen ja asiakkaan näkemykset

Ensin verrataan Samlinkin ja asiakkaiden näkemyksiä toisiinsa. Aiheina olivat yrityksen näkökulmasta ketterän Scrum -sovelluskehitysmenetelmän käyttö, yrityksen odotukset Scrumille, miten yritys aikoo Scrumin avulla vastata asiakkaiden vaatimuksiin ja tarpeisiin sekä syyt siihen, miksi ketterää Scrum -menetelmää päätettiin kokeilla entisen vesiputousmalliin perustuvan systeemyömenetelmän sijaan.

Asiakkaiden näkökulmasta aiheina olivat odotukset ketterälle sovelluskehitykselle ja Scrumille sekä aikaisemman vesiputousmalliin perustuvan systeemyömenetelmän hyvät ja huonot puolet.

Yrityksen näkökulmaa ryhdyttiin selvittämään Samlinkin asiakastyytyväisyystutkimuksen avulla. Sovelluspalvelut-yksikön johtaja Simola (2011) kertoi, että asiakastyytyväisyystutkimuksen mukaan asiakkaat pitivät hyvänä Samlinkin luotettavuutta, kotimaisuutta, asiantuntevuutta ja osaamista. Lisäksi he arvostivat Samlinkia pitkäaikaisena toimittajana. Kehittämiskohteiksi asiakkaat olivat nostaneet jäykkyyden, hitauden, joustamattomuuden ja kalleuden. Asiakkaat olivat myös toivoneet, että heitä kuunneltaisiin enemmän ja, että sovelluskehitystyö olisi enemmän asiakas- ja käyttäjälähtöistä (Samlinkin asiakastyytyväisyystutkimus 2010). Asiakkaiden esittämiä kehittämiskohteita tukevat myös asiakkaiden näkemykset aikaisemmasta systeemyömenetelmästä. Se on heidän mielestään erittäin kankea, joustamaton ja jäykkä, etenkin muutoksia haluttaessa. Asiakkaat näkevät ongelmia myös projektien aikatauluissa, eli sovelluskehitys on kestänyt liian kauan ja näin myös lopulliset kustannukset ovat kasvaneet alkuperäisistä arvioista (Asiakaskysely 2011).

Scrum-menetelmän kokeileminen oli asiakkaiden toivomus. Asiakkaat halusivat kokeilla ketterämpää sovelluskehitystapaa ja olivat esittäneet Scrumin kokeilemista. Useat yritykset olivat

käyneet esittelemässä ketterää Scrum -menetelmää Samlinkille. Esittelyissä sekä Scrumiin liittyvistä aineistoista oli selvinnyt, että Scrumia on käytetty monissa samankaltaisissa projekteissa kuin Samlinkissa toteutettavat sovelluskehitysprojektit. Koska yrityksessä käynnistyi uusi sähköisen työpöydän pilottiprojekti, päätettiin kyseisessä projektissa kokeilla ketterän Scrum-menetelmän käyttöä (Simola 2011).

Samlink odottaa ketterän Scrum - menetelmän ja Scrum-tiimien parantavan joustavuutta, tuottavuutta sekä henkilöstötyytyväisyyttä. Näin Scrum tukee erittäin hyvin myös yrityksen strategiaa. Samlink toivoo Scrum-menetelmän avulla tuottavansa asiakkailleen mahdollisimman suurta liiketoiminta-arvoa, parantavansa asiakastytyväisyyttä sekä asiakas- ja käyttäjälähtöisyyttä. Lisäksi, koska asiakastytyväisyystutkimuksen mukaan asiakkaat kokevat yrityksen kehittämiskohteiksi joustamattomuuden, kankeuden ja hitauden, yrityksen täytyy löytää menetelmiä ja keinoja, joiden avulla kehitystyötä pystytään tekemään joustavammin ja nopeammin ja näin mahdollistamaan tulevaisuudessa tuotoksien nopeampi saaminen markkinoille (Simola 2011).

Asiakkaat odottavat ketterän Scrum - menetelmän tuovan avoimempaa keskustelua, tehokkaampaa yhteistyötä sekä joustavampaa ja nopeampaa sovelluskehitys- ja muutosprosessia (Asiakaskysely 2011). Yrityksen ja asiakkaiden odotukset kohtaavat hyvin ja ne toteutuvat siten, että asiakkaat Scrumin mukaisesti osallistuvat koko ajan kehitystyöhön, näkevät tuotoksen tietyn väliajoin ja pysyvät siksi jatkuvasti ajan tasalla kehitystyön etenemisessä. Lisäksi he ovat koko ajan mukana päättämässä projektin etenemisestä sekä määrittelemässä ja päättämässä yhdessä Samlinkin kanssa vaatimuksista.

Aikaisemmassa systeemityömallissa Simola (2011) näki suurimpina ongelmina sen, että asiakas osallistuu kehitystyöhön hyvin vähän ja kestää kauan ennen kuin asiakas näkee tuotoksen. Myös asiakkaat näkivät tämän ongelmana, koska heidän mukaansa määrittelyvaihe, johon he yleensä osallistuivat, on usein erittäin raskas ja kestää kauan. Näin lopullisen tuotoksen näkemiseen kuuluu pitkä aika. Asiakkaat myös kokivat ongelmaksi sen, että ison kokonaisuuden suunnitteleminen täysin valmiiksi jo määrittelyvaiheessa on mahdotonta. (Asiakaskysely 2011). Ketterän Scrum -menetelmän pitäisi tuoda edellisiin ongelmiin parannusta, koska sitä noudattamalla Sprinttien aikana muutoksia voidaan tehdä joustavammin ja nopeammin. Työtä tehdään pienissä paloissa, joten isoja kokonaisuuksia ei jouduta heti määrittelemään. Scrumin pitäisi myös parantaa asiakkaiden kokemia aikataulu- ja kustannusongelmia, koska sen mukaan toimittaessa projektin sisältö joustaa, mutta yleensä aikataulu ja kustannukset pysyvät muuttumattomina. Asiakkaat myös päättävät yrityksen kanssa, mitä Sprinteissä tehdään, joten heidän päätöksillään ja valinnoillaan on suuri vaikutus siihen, minkälainen valmiista tuotoksesta tulee.

Seuraavissa luvuissa kuvataan sähköisen työpöydän pilottiprojektiin osallistuneiden muiden asiantuntijoiden näkemyksiä. Näistä näkemyksistä tehtiin yhteenveto. Asiantuntijoiden näkemyksiä vertaillaan myös yrityksen ja asiakkaiden näkemyksiin niissä kohdissa missä se on mahdollista.

4.2 Scrumin suurimmat edut ja hyödyt

Projektiin osallistuneiden asiantuntijoiden mielestä Scrumin kolmeksi suurimmaksi eduksi ja hyödyksi nousivat sovelluskehityksen tekeminen pienissä paloissa, tehtävien pilkkominen sekä Scrum-tiimi.

Sovelluskehityksen tekemisessä pienissä paloissa ja tehtävien pilkkomisessa nähtiin seuraavia etuja ja hyötyjä.

Tuloksia saadaan nopeasti nähtäville, asiakkaat osallistuvat määrittelyyn ja näin ollen pystyvät vaikuttamaan muutoksiin jo Sprinttien alkuvaiheessa. Toteutuksen etenemistä on helppompi seurata ja toteutustyötä saa tehdä monipuolisesti ja siten yleisymmärrys asioista kasvaa ja jakaantuu tiimin kesken. Tehtävien pilkkominen pienemmiksi helpottaa toteutustyötä, koska ensin voidaan toteuttaa ne toiminnallisuudet, joilla on liiketoiminta-arvoa. Tehtävien pilkkomisen vuoksi työn tekeminen on myös mielekkäämpää ja tehtävät tulee tarkemmin suunniteltua. Koska kyseessä oli ensimmäinen Scrum -menetelmän pilottikokeilu, oli vertailutietoa vielä vähän. Vielä ei oikein tiedetty eikä osattu toimia Scrumin mukaan. Tästä huolimatta ketterää Scrum -menetelmää pidettiin kuitenkin hyvänä menetelmänä ja sen uskotaan tuovan paljon etuja ja hyötyjä, kunhan sen käyttö opitaan paremmin (Haastattelututkimus 2011).

Osa vastaajista näki pienissä paloissa tekemisessä ja tehtävien pilkkomisessa myös ongelmia. Työ on hyvin dynaamista ja siksi jonkun täytyy huolehtia siitä, että kokonaisuus pysyy hallinnassa. Asiat täytyy osata pilkkoa sopiviin osiin ja tarkkaan mieltä, mitkä osat voidaan toteuttaa irrallaan toisistaan, ettei synny sellaisia osia, joita ei voida yhdistää mihinkään. Tehtävien pilkkomisessa koettiin alussa myös haasteita. Tehtävien kokoluokkaa ei osattu arvioida oikein, jolloin niitä ei pilkottu tarpeeksi pieneksi ja näin ollen niiden työmäärän arvioiminenkin oli hankalaa (Haastattelututkimus 2011).

Scrum-tiimin etuina ja hyötyinä nähtiin seuraavia asioita.

Tiimityöskentelyä pidettiin parempana vaihtoehtona kuin yksintyöskentelyä, koska tiimityö ja yhdessä tekeminen tehostavat työskentelyä. Lisäksi yhteisessä tilassa työskentely helpottaa ja edistää kommunikointia. Tärkeänä pidettiin myös sitä, että tiimi saa itse arvioida työmäärät sekä sitä, että testaaja on koko ajan mukana tiimin jäsenenä. Tiimillä on myös katselmointi-

käytäntö, jonka avulla pyritään kasvattamaan ja jakamaan asioiden ja töiden yleisymmärrystä. Katselmointi tehdään kun työ on valmis. Tiimin jäsen esittelee työn toiselle tiimin jäsenelle, minkä jälkeen se katselmoidaan yhdessä (Haastattelututkimus 2011).

Asiantuntijoiden näkemysten perusteella voidaan päätellä, että yrityksen odottamaa henkilöstötyytyväisyyden paranemista voidaan ajan mittaan saavuttaa Scrum-tiimien avulla.

Scrum-tiimin heikkouksina nähtiin, että osa tehtävistä jaetaan vielä sen mukaan, kuka niitä on aikaisemmin tehnyt tai kuka niistä tietää parhaiten, vaikka Scrumin mukaan tiimin ideana on juuri tehtävien jakaminen. Osaksi näin on käynyt sen vuoksi, että tehtäviä on ollut vielä vähän (Haastattelututkimus 2011).

Projektissa mukana olevilta asiantuntijoilta kysyttiin myös mielipiteitä yhteistyöstä Jyväskylän tiimin kanssa. Samlinkin Espoon tiimin lisäksi sähköisen työpöydän pilottiprojektissa on työskennellyt myös toinen tiimi Jyväskylässä. Kokemuksia yhteistyön sujumisesta kysyttiin Espoon tiimiltä, koska haluttiin tietää, miten eri paikkakunnilla työskentely vaikuttaa siihen.

Yhteistyö ja kommunikointi Jyväskylän ja Espoon tiimin välillä koettiin hieman hankalaksi, kun ollaan eri paikkakunnilla. Eri kommunikaatiovälineet, kuten sisäinen IRQ-kanava, Office Communicator ja web-kamerat ovat kuitenkin parantaneet sitä. Tärkeänä pidettiin sitä, että tiimit tekevät omia toiminnallisia kokonaisuuksiaan. Näin ollen välttyään ongelmilta, kuten päällekkäisyyksiltä ja epäselvyyksiltä. Ongelmana pidettiin sitä, että tiimit eivät vielä tarpeeksi hyvin tiedä mitä kukin tiimi tekee (Haastattelututkimus 2011).

4.3 Scrum vs. perinteinen menetelmä

Scrumin avulla saa selkeän kuvan siitä, miten kehitystyötä tehdään iteratiivisesti ja inkrementaalisesti. Tiimityöskentely koettiin paremmaksi vaihtoehdoksi kuin yksintyöskentely, koska tiimin kanssa työt voidaan jakaa. Tärkeänä pidetään työn parempaa seurantaa, asiakkaiden jatkuvaa osallistumista, ja sitä, että tiimi ja asiakkaat tietävät koko ajan miten kehitys edistyy ja mitä on vielä tekemättä. Työ koetaan myös vaihtelevammaksi ja kokonaisuus pystytään paremmin hahmottamaan. Tärkeänä pidetään myös sitä, että muutoksiin ja mahdollisiin aika-tilaongelmiin pystytään reagoimaan joustavammin ja nopeammin, samoin mahdolliset esteet ja hidasteet havaitaan aiemmin. Lisäksi dokumentaation vähentyminen nähdään etuna, koska tällöin jää enemmän aikaa itse toteuttamiseen (Haastattelututkimus 2011).

Projektissa mukana olleiden asiantuntijoiden näkemykset kohtaavat hyvin myös yrityksen ja asiakkaiden näkemykset. Yritys odottaa joustavuutta (Simola 2011) ja asiakkaat toivovat sujuvampaa kehitys- ja muutosprosessia (Asiakaskysely 2011.) Näiden on mahdollista toteutua si-

ten, että muutoksiin pystytään vastaamaan joustavammin ja nopeammin (Haastattelututkimus 2011.) Asiakkaat toivovat myös projektin joustavaa etenemistä sekä aikataulujen pitävyyttä (Asiakaskysely 2011.) Nämä voivat toteutua siten, että töitä pystytään seuraamaan paremmin, ja koko ajan tiedetään miten projekti etenee. Myös aikataulujen pitäminen on helpompaa, koska mahdollisiin aikatauluongelmiin pystytään reagoimaan aikaisemmassa vaiheessa (Haastattelututkimus 2011.)

4.4 Sprint Planning - tapaaminen

Sprint Planning -tapaaminen kestää päivän ja siihen osallistuvat tiimien jäsenet. Asiantuntijoilta kertoivat ensin mitä tapaamisissa tehdään.

Sprint Planning -tapaamisessa arvioidaan Scrumin mukaisesti, paljonko asioita voidaan Product Backlogista ottaa seuraavaan Sprinttiin. Product Owner on käynyt edellisenä päivänä asiakkaiden kanssa läpi asiakkaiden toivoman toteutusjärjestyksen. Tämän jälkeen valitut asiat pilkotaan tehtäviksi ja niiden työmäärät arvioidaan. Työmääräarvioinnissa käytetään ”Planning Poker” käytäntöä, missä jokainen antaa oman arvionsa tehtävälle ja niistä lasketaan keskiarvo. Lisäksi päätetään yhdessä Sprintille tavoite, joka kirjataan valkotauluun (Haastattelututkimus 2011).

Sprint Planning -tapaamisesta nousi esiin muun muassa, että projektin alussa tehtävät pilkottiin liian suuriksi kokonaisuuksiksi, jolloin työmäärien arvioiminen oli hankalaa. Nyt tilanne on parantunut, kun tehtävät on opittu pilkkomaan pienemmiksi. Tärkeänä pidetään ulkopuolisen mentorin apua, koska tämä neuvoi, mikä tapaamisen tarkoitus on ja miten siellä pitää toimia (Haastattelututkimus 2011).

4.5 Product Backlog

Product Backlogia koskevista vastauksista kävi ilmi, että vaikka Product Owner vastaa Product Backlogista, sitä ovat välillä ylläpitäneet muutkin kuten, Scrum Master tai projektipäällikkö. Product Backlog on jollakin tavalla priorisoitu, mutta ei vielä parhaalla mahdollisella tavalla. Projektin alussa oli ongelmia, koska asiat olivat liian suuria kokonaisuuksia. Product Backlog tuntui sekavalta ja asiat epämääräisiltä eikä Product Backlogin asioita osattu pilkkoa tarpeeksi pieniksi. Pilkkomista on nyt opittu, joten tilanne on parantunut. Ylimpänä ovat korkeimman prioriteetin asiat eli ne, jotka täytyy tehdä seuraavaksi. Tiimiläiset eivät aina ole tienneet, missä Product Backlog sijaitsee, ja se on jäänyt hieman etäiseksi (Haastattelututkimus 2011).

4.6 Sprintit

Sprinttejä koskevista vastauksista kävi ilmi, että sitä pidettiin tärkeänä, että Sprintillä on tavoitteet ja niiden on oltava selkeitä. Koettiin, että projektin alussa oli vaikea arvioida, kuinka paljon toteutusta pystytään tekemään yhden Sprintin aikana. Sprinttien aikana ei ole yritetty lisätä tehtäviä Sprinttiin, vaan ennemminkin siitä oli jouduttu poistamaan tehtäviä, koska niitä ei ehditty tekemään tai niissä oli epäselvyyksiä, kuten teknisiä ongelmia. Erittäin tärkeänä pidettiin myös sitä, että tiimin jäsen on mukana vain tiimissä eikä useassa projektissa, jotta työhön saa keskittyä ja sitä saa tehdä rauhassa. Resursoinnin kanssa koettiin siis vielä ongelmia ja toivottiin resursointiin ryhtiä (Haastattelututkimus 2011).

4.6.1 Sprint Backlog

Projektissa Sprint Backlogina käytetään valkotaulu-käytäntöä, mikä tarkoittaa, että Sprintin tehtävät ja niiden työmääräarviot ovat taululla nähtävissä. Taululla on omat sarakkeet aloittamattomille töille, tekeillä oleville töille ja valmiille töille.

Vastaajat pitivät tärkeänä sitä, että ulkopuolinen mentori opasti ja neuvoi, miten valkotaulu-käytäntö toimii. Valkotaulun käyttö koettiin selkeäksi, koska sen avulla nähdään, missä ollaan menossa. Projektin alussa tehtävien kokoluokan kanssa oli ongelmia, eli alussa tehtävät olivat liian suuria ja näin ollen myös niiden työmääriä oli vaikea arvioida. Nyt tilanne parantunut, kun tehtävät on pilkottu pienemmiksi (Haastattelututkimus 2011).

4.7 Product Owner

Product Owner on toiminut yhteyshenkilönä asiakas-Product Ownereiden - tiimissä. Lisäksi hän on vastannut Product Backlogista, sähköisestä työpöydästä, konseptoinnista ja kokonaisnäkemyksestä. Product Ownerin rooli, eli vastuut ja tehtävät, on ollut tiimin muille jäsenille hieman epäselvä, mutta rooli on ollut epäselvä myös Product Ownerille itselleen. Yhteistyö tiimin ja Product Ownerin välillä on kuitenkin sujunut hyvin. Sekä tiimin muut jäsenet että Product Owner näkevät Product Ownerin roolin tärkeänä. Product Ownerin täytyy koko ajan osallistua kehitykseen tuote-alueen asiantuntijana. Product Ownerilla täytyy myös olla valtaa tehdä päätöksiä ja hänen täytyy myös tehdä niitä. Tärkeänä pidettiin myös sitä, että Product Owner tietää, mitä asiakkaat haluavat, koska hän toimii linkkinä asiakkaiden ja tiimin välillä (Haastattelututkimus 2011).

Product Owner on yrittänyt hallita kokonaisuutta ja jatkuvasti miettiä, miten sähköiselle työpöydälle voidaan tuoda uusia toiminnallisuuksia. Alussa ehkä oletettiin, että yksi Product Owner pystyisi tuntemaan kaikki liiketoiminta-alueet ja vastamaan niistä. Jatkossa on tarkoitus, että jokaiselta liiketoiminta-alueelta saadaan oma Product Owner.

4.8 Daily Scrum ja Scrum Master

Daily Scrum -tapaamiset ovat pääasiallisesti toimineet hyvin. Ulkopuolisen mentorin mukana olo koettiin tärkeäksi avuksi Daily Scrum-käytännön omaksumisessa. Mentori oli alussa mukana neuvomassa, miten Daily Scrummeissa kuuluu toimia. Daily Scrumit järjesti Scrum Master ja ne niiden kesto pyrittiin rajoittamaan 15 minuuttiin, mitä osanottajat pitivät tärkeänä. Tämä edellyttää Scrum Masterilta jämäkkyyttä. Daily Scrummeissa oli pääasiallisesti keskitytty Scrumin mukaisesti kolmeen kysymykseen (Haastattelututkimus 2011). Mitä olet tehnyt viime palaverin jälkeen? Mitä aiot tehdä ennen huomista Daily Scrumia? Estääkö joku tai jokin sinua tekemästä työtä niin tehokkaasti kuin mahdollista? (Schwaber & Beedle 2002, 43; Settler 2011, 90.)

Asiantuntijoiden vastausten perusteella Sprinteissä ei törmätty juurikaan esteisiin, koska tehtäviä oli vielä vähän. Scrum Master oli poistanut osan esteistä ja projektipäällikkö osan. Ilmeni myös, että tiimille ei ollut täysin selvää, mistä esteistä sen pitäisi kertoa Scrum Masterille. Pilottiprojektin Scrum Master ei ollut tekninen asiantuntija, minkä vuoksi hän ei voinut juurikaan poistaa teknisiä esteitä. Tämän takia useat asiantuntijat miettivätkin haastattelussa sitä, pitäisikö Scrum Masterilla olla enemmän teknistä osaamista (Haastattelututkimus 2011).

4.9 Sprint Review -tapaaminen

Sprint Review -tapaaminen kestää päivän ja sisältää asiakas-Product Owner -tapaamisen, demotilaisuuden sekä Retrospective -tapaamisen.

Sprint Review -tapaamisista nousi esiin muun muassa, että alussa oli epäselvää, kuka tapaamisia vetää, miten päivän on tarkoitus kulkea jne. Nyt tilanne on parantunut ja tapaamiset sujuvat jo melko hyvin. Alussa olisi myös tarvittu enemmän ulkopuolisen mentorin apua. Mentoria olisi tarvittu kertomaan, miten tapaamisissa toimitaan, mitä tehdään ja miksi. Tapaamisiin osallistui monen asiantuntijan mielestä välillä liian monta henkilöä ja ne kestävät liian kauan. Jatkossa toivotaan, että tapaamisia voidaan tiivistää. Lisäksi tapaaminen edellyttää Product Ownerilta paljon selvittelyä, määrittelyä ja valmistelua (Haastattelututkimus 2011).

Asiakas PO -tapaamisiin osallistuvat Samlinkin Product Owner ja asiakkaiden Product Ownerit.

Asiakkailta kysyttiin mitä hyvää ja mitä huonoa heidän mielestään on ollut asiakas Product Owner -tapaamissa.

Asiakkaiden mielestä on tärkeää, että tapaamisiin osallistuu eri pankkiryhmien edustajia ja että hyvin hallittavien asiakaskokonaisuuksin läpikäynti on ollut mielekästä. Tärkeäksi nähtiin myös, että yhteisestä kehittämisestä pystytään keskustelemaan avoimesti ja mahdollisuus jatkuvaan kommentointiin mahdollistaa muutoksien toteuttamisen varhaisessa vaiheessa. Hyvänä pidettiin myös sitä, että ryhmän ammattiosaaminen on laaja-alaista. Osallistujat ovat myös valmistautuneet palavereihin etukäteen ja näin ollen asioita saadaan helpommin edistettyä. Lisäksi koettiin, että Samlinkin Product Owner on hoitanut oman osuutensa hyvin (Asiakaskysely 2011).

Asiakkaat kokivat, että kokousaineisto ei aina ollut ajan tasalla, ja joskus se saatiin niin myöhään, että osallistujat eivät ehtineet saamaan kommentteja oman pankkiryhmänsä edustajilta. Eritasoisien asiantuntijoiden mukana olo oli joskus koettu hankalaksi, kun käytetään eri alueilla käytössä olevaa ”slangia” (Asiakaskysely 2011).

Demotilaisuuteen osallistuvat tiimien jäsenet, asiakkaat sekä muita Samlinkin henkilöitä.

Demotilaisuudesta nousi muiden asiantuntijoiden eli tiimin jäsenten mukaan esiin seuraavia asioita.

Tilaisuudet ovat sujuneet hyvin ja asiakkaat ovat antaneet kommentteja sekä kertoneet mielipiteitään tuotoksesta ja siihen tehdyistä lisäyksistä. Näin tiimi on saanut hyödyllistä palautetta. Tilaisuuksista on jäänyt sellainen tunne, että asiakkaat ovat olleet tyytyväisiä, kun näkevät tuotoksen ja siihen tehdyt lisäykset, ja näkevät, missä kehityksessä ollaan menossa. Koettiin myös, että kaikkien tiimin jäsenten ei välttämättä tarvitsisi olla aina paikalla (Haastattelututkimus 2011).

Asiakkaiden näkemykset kohtaavat hyvin muiden asiantuntijoiden näkemykset, sillä asiakkaat pitivät hyvänä kehittäjien ja asiakkaiden välistä avointa keskustelua. Myös tuotoksen nopea tuominen heidän nähtäväkseen ja arvioitavakseen on tärkeää. Tärkeää asiakkaiden mielestä on myös se, että he tietävät, miten kehitys edistyy. Lisäksi asiakkaat näkivät hyvänä sen, että muutokset kirjataan nopeasti muistiin ja ne saadaan nopeasti toteutettavaksi. Huonoina asioina asiakkaat pitivät demojen esittelyä silloin, kun ne eivät sisällä uutta toiminnallisuutta sekä sitä, että pieniä yksityiskohtia käsitellään joskus turhan kauan. Lisäksi joskus toivotaan, että demotilaisuuteen voisi osallistua useampi henkilö pankkiryhmästä (Asiakaskysely 2011).

4.9.1 Retrospective

Retrospectiveen osallistuvat tiimien jäsenet ja joskus mukana on myös muita Samlinkin henkilöitä, kuten menetelmävastuuhenkilöt ja muita projektiryhmäläisiä. Asiantuntijat kertoivat ensin mitä tapaamisessa tehdään.

Retrospective -tapaamisessa käydään Scrumin mukaisesti läpi mitä edellisessä Sprintissä meni hyvin, mikä huonosti ja mitä voitaisiin parantaa. Lisäksi selvitetään onko edellisten Sprinttien havainnoille tehty mitään ja miten on onnistuttu Sprintin tavoitteisiin nähden. Lopuksi on käyty läpi myös sitä, mitä seuraavassa Sprintissä voitaisiin ryhtyä tekemään sekä tehty valmisteluja seuraavan päivän Sprint Planning -tapaamista varten (Haastattelututkimus 2011).

Retrospective -tapaamisesta nousi esiin muun muassa, että projektin alussa oli vähän sekavaa, kun ei tiedetty, miten tapaamisen pitäisi edetä ja mitä siellä pitäisi tehdä. Tilanne on vähitellen parantunut, kun on opittu lisää. Tapaamiset vaativat kuitenkin edelleen tehostamista. Koettiin, että tapaamisissa on välillä liian monta osallistujaa. Koettiin myös, että sekä Samlinkin että Jyväskylän tiimin ei tarvitse välttämättä aina osallistua samaan tilaisuuteen. Toisaalta molempien tiimien läsnäolo tuo lisäarvoa, kun tiimit vaihtavat ajatuksia keskenään. Lisäksi jatkossa pitäisi enemmän paneutua niihin asioihin, joita on käyty läpi, jotta asioita saataisiin parannettua (Haastattelututkimus 2011).

4.10 Organisaatio

Haastatteluissa vastaajilta kysyttiin, tukeeko organisaatio heidän mielestään Scrumin käyttöä. Osa vastauksista liittyy projektinhallinnallisiin asioihin mutta, on tärkeää kertoa myös ne.

Organisaatio - positiivisia havaintoja

Vastaajat kokivat, että organisaatio kannustaa ketterään työskentelyyn, koska yhteisen tilan saamiseen oli suhtauduttu positiivisesti ja osanottajille on annettu esimerkiksi työrauha. Lisäksi projektiin saatiin ulkopuolisia mentoreita neuvomaan ja tukemaan.

Organisaatio - muita havaintoja

Vastaajat totesivat, että perehtymiselle ja oppimiselle täytyy varata riittävästi aikaa. Heistä tuntui, että johto odottaa ketterän sovelluskehityksen mahdollistavan tuotoksien valmistumisen paljon nopeammin kuin aikaisemman sovelluskehitysmallin aikana. Haasteeksi koettiin projektin sidokset moneen muuhun projektiin, koska toisista projekteista ei oikein saa tietoa ja lisäksi jonkun pitäisi myös hallita projektien kokonaisuutta. Lisäksi muut projektit eivät vielä tue ketterää toimintaa. Vastaajat kokivat, että tekniseen tietämykseen olisi pitänyt heti alussa saada tukea. Resursseja täytyisi varata etukäteen ja resurssien pitäisi myös olla tiimin käytössä, koska muuten työt viivästyvät. Tiimin jäsenillä ei saa olla muita projekteja samanaikaisesti, vaan tiimin jäsenten täytyy saada keskittyä yhteen projektiin kerrallaan (Haastattelututkimus 2011).

4.11 Muut huomioitavat asiat

Projektiin osallistuvilta asiantuntijoilta kysyttiin myös millaisia asioita he pitivät epäselvinä tai mitä muita havaintoja he tekivät joko ketterästä Scrum -menetelmästä tai sen käytännön soveltamisesta sähköisen työpöydän pilottiprojektissa.

Scrumiin liittyviä epäselvyyksiä ja havaintoja:

Tutkimuksen kohteena ollut projekti oli Scrum - menetelmän pilotti, eikä siinä vielä oikein osattu toimia Scrumin mukaisesti. Scrumin käytännöt ja Scrum Masterin ja Product Ownerin roolit sekä niiden vastuut ja tehtävät olivat vähän epäselviä. Projektissa oli mukana myös projektipäällikkö ja näiden kolmen eri roolin vastuut ja tehtävät jäivät vielä epäselviksi. Ei oikein tiedetty, mistä asioista kukakin vastaa. Lisäksi Scrumin mukaiselle dokumentaatiolle asetettuja vaatimuksia ei vielä täysin ymmärretty eikä tiedetty, millaista dokumentaatiota olisi pitänyt tehdä (Haastattelututkimus 2011).

Projektinhallinnassakin oli asioita, jotka koettiin epäselviksi tai niistä tehtiin muita havaintoja. Tärkeää oli kertoa myös näistä.

Projekti olisi kaivannut enemmän teknistä tukea ja tekninen suunnittelu olisi pitänyt ottaa huomioon paremmin. Ympäristöjen kanssa oli ongelmia, mikä vaikeutti esimerkiksi testausta. Tarvittavien asiantuntijoiden mukaan saaminen oli välillä hankalaa ja asioiden suunnittelu jäi liian vähäiseksi. Dokumentaatiota oli useassa eri paikassa, se oli välillä epäselvää eikä kaikkia dokumentteja ylläpidetty. Lisäksi ongelmia aiheutui siitä, että toteutusta tehtiin aikaisemman vesiputousmalliin perustuvan menetelmän määrittelyjen perusteella, jolloin testattavaa saatiin vasta myöhäisessä vaiheessa (Haastattelututkimus 2011).

4.12 Kehittämideoita ja havaintoja

Haastateltavilta kysyttiin, mitä heidän mielestään pitäisi jatkossa kehittää ja/tai parantaa tai erityisesti ottaa huomioon, jotta tulevien projektien olisi mahdollisimman helppoa ottaa ketterä Scrum - menetelmä käyttöön. Osa vastauksista liittyy projektinhallinnallisiin asioihin.

Vastausten perusteella tärkeimpänä asiana pidetään ulkopuolisen avun saamista tehtävien pilkkomiseen ja niiden arvioimiseen. Tämä oli melkein kaikkien vastaajien mielestä tärkein havainto. Tärkeänä pidettiin myös, että ainakin alussa pitäisi olla mukana ulkopuolinen mentor, joka kertoo, mitä Scrumin käytännöt tarkoittavat sekä miten tapaamissa toimitaan ja miksi. Myös Scrum Masterilla ja Product Ownerilla pitäisi olla ulkopuoliset mentorit. Koulutus-

ta tarvitaan käyttäjätarinoiden ja skenaarioiden tekemiseen, jotta varmistetaan niiden laadukkuus. Koulutusta tarvitaan myös työvälineiden käyttöön. Lisäksi Scrumin käytäntöjen mu-
kaisten tapaamispäivien ajankohta pitäisi miettiä jo hyvissä ajoin ennen ensimmäisen Sprintin
alkamista, koska niiden vaihtaminen on lähes mahdotonta.

Tärkeänä pidettiin myös sitä, että tiimi saa työskennellä yhdessä tilassa ja että testaaaja on
koko ajan tiimissä mukana. Tiimin täytyy tietää, mitä dokumentaatiota projektin aikana täy-
tyy tehdä ja varmistaa, että dokumentaatio löytyy yhdestä paikasta ja että sitä ylläpidetään.
Tarvittavien sidosryhmien henkilöiden täytyy olla resursoitu projektiin ja mieluummin olla osa
tiimiä. Lisäksi koetaan, että yrityksen ketterän kehitysryhmän pitäisi olla enemmän mukana
projekteissa näkemässä ja avustamassa käytännön tekemistä, jotta ymmärrys ketterän mu-
kaan toimimisesta ja ketterän menetelmän ohjeistuksista saadaan koko talotasolla jalkautet-
tua (Haastattelututkimus 2011).

Lopuksi haastateltavilta kysyttiin, tekisivätkö he mieluummin töitä ketterän Scrum - mene-
telmän kuin aikaisemman perinteisen menetelmän mukaan. Vastausten mukiaan enemmistö
työskentelisi mieluummin Scrumin mukaisesti. Perusteluja olivat muun muassa asioiden teke-
minen monipuolisemmin, kevyempi dokumentaatiomalli ja koettiin myös, että tiimin tuki te-
hostaa työskentelyä. Lisäksi koettiin Sprinttien mukainen työskentely mielekkääksi ja vastaa-
jista tuntui, että se tukee varsinkin uuden tekemistä.

5 Yhteenveto

5.1 Vastaaminen tutkimusongelmaan ja tutkimuskysymyksiin

Ensisijaisena tutkimusongelmana oli selvittää, miten sähköisen työpöydän pilottiprojektissa
mukana olleet asiakkaat ja muut asiantuntijat ovat kokeneet ketterän Scrum - menetelmän
käytön, ja millaisia kehittämis- ja parannusehdotuksia heillä on, jotta ketterän Scrum - mene-
telmän käyttöönotto ja käyttö olisi tulevissa Samlinkin sovelluskehitysprojekteissa mahdolli-
simman helppoa ja joustavaa. Näistä syntyivät myös tutkimuksen tutkimuskysymykset.

Samalla haluttiin selvittää, mitä Samlink yrityksenä ja sen asiakkaat odottavat ketterältä
Scrum - menetelmältä, miksi ketterää Scrum - menetelmää päätettiin kokeilla sekä miten yri-
tyksen, asiakkaiden ja muiden asiantuntijoiden näkemykset kohtasivat toisensa.

Lisäksi täytyi selvittää ja ymmärtää Scrumin ideologia, sen käyttäminen sovelluskehityspro-
jektissa sekä sen ero perinteiseen vesiputousmalliseen menetelmään. Tämän avulla voitiin
määritellä, minkälaisia asioita haastateltavilta piti kysyä ja minkälaista teoreettista aineistoa
piti hankkia tukemaan näitä näkemyksiä.

5.1.1 Vastaaminen tutkimusongelmaan

Tutkimusongelmaan löytyi hyvin vastauksia. Ketterän Scrum-menetelmän käytön kokeilu oli lähtenyt asiakkaista koska, heidän mielestään Samlinkin pitäisi kehittää toimintojaan, jotta sen toiminta ei olisi niin jäykkää, hidasta ja joustamatonta. Lisäksi asiakkaat kokivat sovelluskehityksessä aikaisemmin käytetyn vesiputousmalliin perustuvan menetelmän jäykkänä, hitaana ja joustamattomana. Samlinkin täytyi yrityksenä vastata asiakkaiden tarpeisiin ja vaatimuksiin ja siksi yritys kannatti ketterän Scrum-menetelmän käyttöä ja toivoi, että Scrumin avulla sovelluskehitystyötä voidaan tehdä joustavammin ja nopeammin, ja näin saada tuotteet nopeammin markkinoille.

Yrityksen ja asiakkaiden Scrum -menetelmälle asettamat odotukset kohtasivat hyvin. Molemmat odottavat joustavampaa ja sujuvampaa sovelluskehitysprosessia aikaisempaan verrattuna. Yritys odottaa lisäksi tuottavansa Scrumin avulla asiakkailleen suurinta mahdollista liiketoiminta-arvoa sekä parantavansa asiakastyytyväisyyttä ja asiakas- ja käyttäjälähtöisyyttä. Asiakkaat taas odottavat lisäksi avointa keskustelua, tehokasta yhteistyötä ja joustavampaa ja nopeampaa muutosprosessia. Molempien odotuksia yritetään saavuttaa siten, että asiakkaat osallistuvat jatkuvasti kehitystyöhön, näkevät tuotoksen ja siihen tehdyt lisäykset tietyin väliajoin ja näin pystyvät koko ajan seuraamaan kehitystyön etenemistä. Asiakkaat myös seuraavat koko ajan projektin etenemisestä ja tekevät etenemiseen liittyviä päätöksiä. Lisäksi he määrittelevät ja päättävät yhdessä Samlinkin kanssa vaatimuksista jolloin muutosprosessi on joustavampaa ja nopeampaa.

Teoreettisen taustan avulla sai selkeän ja riittävän kuvan ketterästä Scrum - menetelmästä ja pystyttiin määrittelemään aihealueet ja kysymykset haastatteluja ja kyselyä varten. Haastattelujen ja kyselyn aikana selvisi myös, millaista teoriaa piti hankkia näiden näkemysten tukemiseksi.

5.1.2 Vastaaminen tutkimuskysymyksiin

Ensimmäinen kysymys oli, miten Scrumin pilottikäyttö onnistui Samlinkin sähköisen työpöydän pilottiprojektissa.

Vastausten perusteella ketterän Scrum - menetelmän käyttöönotto ja käyttö onnistui hyvin, vaikka kyseessä oli menetelmän pilottiprojekti. Kaikki vastaajat pitivät hyvänä ketterää menetelmää, jossa sovelluksia kehitetään vähittäin ja jossa tulokset nähdään tietyin väliajoin. Kaikki myös tekisivät sovelluskehitystä mieluummin ketterän Scrum-menetelmän mukaisesti kuin aikaisemman vesiputousmalliin perustuvan menetelmän mukaisesti. Asiantuntijoiden

perusteluja tälle olivat muun muassa asioiden monipuolinen tekeminen, kevyempi dokumentaatio, Sprinttien mukainen työskentely, varsinkin uuden tekemisen tukeminen sekä tehokkaampi työskentely tiimin tukemana. Koska kyseessä oli pilottikokeilu, vielä ei oikein tiedetty ja osattu toimia Scrumin mukaan, mutta tästä huolimatta asiantuntijat pitivät ketterää Scrum -menetelmää hyvänä ja uskoivat sen tuovan paljon etuja ja hyötyjä, kunhan sen käyttö opitaan paremmin.

Kaikki vastaajat olivat myös yksimielisiä siitä, että asiakkaan osallistuminen kehitystyöhön on erittäin tärkeää. Osallistumisen suurimpina etuina pidettiin sitä, että vaatimuksia määriteltiin ja niistä päätettiin yhdessä sekä sitä, että syntyvä tuotos ja siihen tehtävät lisäykset nähdään säännöllisin väliajoin, jolloin niistä ja mahdollisista muutoksista voidaan keskustella.

Suurimmiksi ketterän Scrum -menetelmän eduiksi ja hyödyiksi nousi asiantuntija-vastaajien mielestä kolme asiaa: sovelluskehityksen tekeminen pienissä osissa, tehtävien pilkkominen sekä Scrum-tiimi.

Vastaajat näkivät sovelluskehitystyön tekemisessä pienissä osissa ja tehtävien pilkkomisessa etuina ja hyötyinä muun muassa, että asiakkaat osallistuvat määrittelyyn, saadaan nopeasti näkyviä tuloksia ja asiakas pystyy vaikuttamaan muutoksiin jo Sprinttien alkuvaiheessa. Scrumissa pystytään toteuttamaan ensin tärkeimmät toiminnallisuudet eli toiminnallisuudet, joilla on liiketoiminta-arvoa. Toteutuksen etenemistä on helpompi seurata ja lisäksi toteutustyö on helpompaa, kun tehdään osissa ja tehtävät on pilkottu pieniksi.

Vastaajat näkivät Scrum-tiimin etuina ja hyötyinä muun muassa, että tiimityöskentely ja yhdessä tekeminen tehostavat työskentelyä ja tiimityöskentely koettiin myös paremmaksi vaihtoehdoksi kuin yksintyöskentely. Lisäksi koettiin, että yhdessä tilassa työskentely myös helpottaa ja edistää kommunikointia.

Erittäin tärkeänä pidettiin myös ulkopuolisen mentorin apua mm. Daily Scrum-, Retrospective- ja Sprint Planning - tapaamisissa sekä Sprint Backlog - käytännön selvittämisessä.

Suurimmiksi ongelmiksi nousivat Scrumin käytäntöihin ja niiden toteuttamiseen liittyviä seikkoja, kuten kokonaisuuden hallinta, tehtävien pilkkominen ja niihin liittyvien työmäärien arviointi, Scrumin roolit sekä niiden vastuut ja tehtävät, Scrumin mukainen dokumentaatio, Sprinttien työmäärien arvioiminen, etenkin alussa, Product Backlogin priorisointi sekä Sprint Review- ja Sprint Planning - tapaamisten kestot.

Tehtävien pilkkominen nähtiin kokonaisuuden hallinnan näkökulmasta ongelmaksi, eli vaikka tehtävät pilkotaan, jonkun täytyy huolehtia siitä, että kokonaisuus pysyy hallinnassa. Asiat täytyy osata pilkkoa sopiviin osiin, jotta ei synny sellaisia irrallisia osia, joita ei voida yhdistää

mihinkään. Tehtävien pilkkominen oli myös aluksi hankalaa, kun ei osattu arvioida niiden koluokkaa eikä osattu pilkkoa niitä tarpeeksi pieniksi. Tällöin myös niiden työmäärien arvioiminen oli hankalaa.

Scrumin roolit ja dokumentaatio koettiin hieman epäselväksi. Lähinnä ei tiedetty Product Ownerin ja Scrum Masterin roolien vastuuta ja tehtäviä. Scrumin mukainen dokumentaatio oli myös hieman epäselvä eli ei tiedetty mitä dokumentaatiota olisi pitänyt tehdä.

Sprinteissä koettiin aluksi hankalaksi arvioida, minkä verran toteutusta pystytään tekemään yhden Sprintin aikana. Työtä jouduttiin vähentämään useasta Sprintistä, koska kaikkea ei ehditty tekemään tai asioissa oli epäselvyyksiä, kuten teknisiä ongelmia. Voi myös olla mahdollista, että työtä jouduttiin vähentämään myös sen takia, että ei osattu vielä toimia niin hyvin Scrumin mukaan. Tätä tukevat myös teoreettiset näkemykset.

Product Backlogin priorisointi koettiin vielä hieman sekavaksi eli Product Backlog on priorisoitu, mutta ei vielä ehkä parhaalla mahdollisella tavalla.

Sprint Review- ja Sprint Planning - tapaamiset koettiin liian raskaiksi. Lisäksi koettiin, että niihin osallistui välillä liian monta henkilöä. Tapaamiset kestävät tällä hetkellä kaksi päivää.

Toisessa kysymyksessä kysyttiin, mitä jatkossa pitäisi kehittää ja parantaa, jotta Scrumin käyttö ja sen käyttöönotto olisi tulevissa Samlinkin sovelluskehitysprojekteissa mahdollisimman helppoa ja joustavaa.

Tärkeimmäksi kehittämisen- ja parantamiskohteiksi Scrumin käytäntöihin liittyen nousi muun muassa, että ainakin alussa täytyisi olla ulkopuolinen mentori auttamassa tehtävien pilkkomisessa ja työmäärien arvioimisessa. Tämä oli melkein kaikkien asiantuntija-vastaajien mielestä tärkein havainto. Alussa täytyisi myös olla ulkopuolinen mentori kertomassa ja opettamassa, mitä Scrumin käytännöt tarkoittavat ja miten niiden mukaan toimitaan ja miksi. Lisäksi koettiin, että myös Scrum Masterilla ja Product Ownerilla täytyisi olla ulkopuoliset mentorit. Koulutusta pitäisi saada siihen, mitkä ovat Scrumin mukaiset roolit ja mitkä ovat roolien vastuut ja tehtävät. Koulutusta tarvitaan myös Scrumin mukaisten käyttäjätarinoiden ja skenaarioiden tekemiseen, jotta varmistetaan niiden laadukkuus. Täytyy myös tietää, mitä dokumentaatiota täytyy laatia. Lisäksi erittäin tärkeänä pidettiin myös yhteistä tilaa.

5.2 Yhteenveto tutkimustuloksista

Eri näkemykset kohtasivat hyvin toisensa. Yritys ja projektissa mukana olleet asiakkaat ja muut asiantuntijat näkivät asiat hyvin samalla tavalla niissä asioissa, joissa vertailua pystyttiin tekemään. Yrityksen ja asiakkaiden odotukset ketterän Scrum-menetelmän suhteen kohtasivat hyvin ja tämä on erittäin tärkeää jatkotyöskentelyn kannalta. Jos näkemykset eivät

olisi kohdanneet, yrityksen ja sen asiakkaiden yhteistyö olisi hankalaa. Myös asiantuntijoiden näkemykset olivat enimmäkseen hyvin samantapaisia eli paljon oli asioita, jotka nousivat esiin useammassa haastattelussa.

Asiantuntijoiden haastatteluista nousi esiin myös projektinhallinnallisia vastauksia, jotka myös lisättiin tuloksiin, koska niistä on varmasti apua Samlinkin tuleviin projekteihin ja miksei jonkun toisenkin yrityksen tarpeisiin.

Saavutettuja tuloksia tukevat hyvin myös teoreettiset näkemykset, koska useimpia tuloksissa saatuja näkemyksiä voidaan tukea tutkimuksessa löydettyjen teoreettisten näkemysten avulla. Tämä on erittäin tärkeää ja se vaikuttaa myös tutkimuksen luotettavuuteen.

5.3 Keskustelua tutkimuksesta

Tutkimuksen tekeminen oli haastavaa, sillä tapaustutkimus ei ollut ennestään tuttu. Ennen opinnäytetyötä tehtiin pilotti-tapaustutkimus, mikä ei kuitenkaan riittänyt tekemään tutkimusta vähemmän haasteelliseksi. Haasteellisinta oli tapaustutkimuksen ymmärtäminen sekä teoreettisen ja empiirisen aineiston analysointi. Teoreettisen aineiston analysointi oli haastavaa, koska piti osata valita ja analysoida niitä lähteitä, joiden avulla pystyttiin tukemaan tuloksia. Empiirisen aineiston analysoinnissa haasteena oli enimmäkseen se, miten ja millä tavalla aineistoa pitäisi analysoida, jotta vertailu ja yhteenveto olisivat mahdollisimman helppoja tehdä. Hyvä analyysitapa oli aineiston jakaminen teemoihin eri värien avulla. Tämä helpotti huomattavasti aineiston analyysia, tuloksien vertailua ja yhteenvetoa.

Tutkimuksen luotettavuuden arvioinnissa käytetään käsitteitä validiteetti ja reliabiliteetti (Kananen 2008, 123; Ruusuvuori ym. 2010, 27 - 28.)

Validiteetti tarkoittaa sitä, että tutkitaan oikeita asioita eli sitä, mitä pitääkin tutkia (Kananen 2008, 123.). Validiteettia voidaan arvioida sen mukaan, onko aineisto valittu oikein, jotta saadaan tutkimuskysymyksiin mahdollisimman hyvä ja osuva vastaus (Ruusuvuori ym. 2010, 27.) Tutkimuksessa käytettiin teorian, yrityksen sekä projektissa mukana olleiden asiakkaiden ja asiantuntijoiden näkemyksiä. Näin varmistettiin, että käytettiin useita eri lähteitä ja oikeanlaista aineistoa ja saatiin vastaukset tutkimuskysymyksiin sekä tutkimusongelmaan.

Vastausten analysointi pyrittiin tekemään analyysistrategian avulla mahdollisimman kattavasti ja järjestelmällisesti, jotta mielenkiintoiset asiat pystyttiin tunnistamaan, niitä pystyttiin ryhmittelemään, vertailemaan ja lopulta yhdistämään ne lopullisiksi tuloksiksi. Tutkimuksen arviointiin vaikuttaa se, onko analyysi tehty laadukkaasti ja järjestelmällisesti ja pystyykö

tutkija vakuuttamaan lukijan siitä, että tulkinta on perusteltu ja että havaintoja on riittävästi testattu (Ruusuvuori ym. 2010, 27.)

Reliabiliteetti tarkoittaa tulosten pysyvyyttä eli toistettavuutta (Kananen 2008, 124.) Täyttä varmuutta ei ole siitä, nousisiko samoja teemoja ja niihin liittyviä asioita esille, jos joku muu tekisi saman tutkimuksen uudelleen. Tämän tutkimuksen perusteella voi kuitenkin päätellä, että ainakin joitakin samoja asioita nousisi esille, koska ne nousivat esiin sekä tässä tutkimuksessa että myös useammassa teoreettisessa näkemyksessä.

5.4 Ehdotuksia jatkotutkimukseksi

Tutkimuksen aikana nousi esiin kolme jatkotutkimusaihetta.

Samlinkin kannalta olisi kiinnostavaa tehdä uusi tutkimus saman tiimin kanssa esimerkiksi vuoden tai kahden kuluttua. Tutkimus toisi yritykselle tietoa siitä, onko tässä tutkimuksessa todettuja tuloksia hyödynnetty ja ovatko ne auttaneet asioiden kehittämisessä. Lisäksi mielenkiintoista olisi tehdä tutkimus jonkun muun Samlinkin tiimin kanssa, kun tiimi on työskennellyt yhtä kauan kuin tähän tutkimukseen osallistuva tiimi. Tällöin nähtäisiin nousevatko samat asiat esille muillakin tiimeillä. Jos tämän tutkimuksen tuloksia sekä kehittämis- ja parannusehdotuksia hyödynnetään yrityksessä, seuraavan tutkimuksen tulokset eivät välttämättä olisi samanlaisia.

Yleiseltä kannalta olisi mielenkiintoista, mikäli joku toinen tutkija tekisi tutkimuksen ketterän Scrum-menetelmän käyttöönotosta toisessa yrityksessä. Tällöin nähtäisiin, minkälaisia asioita siinä tutkimuksessa nousisi esille. Tutkimusten avulla saataisiin lisää kokemuksia ketterän Scrum - menetelmän käyttöönotosta ja siitä mitä käytönottovaiheessa pitäisi huomioida.

Toisena tutkimuksena voitaisiin tehdä tutkimus ketterän Scrum - menetelmän käytöstä yrityksen sovelluskehityksessä kun sitä on käytetty vähintään kahden vuoden ajan. Tällä tutkimuksella saataisiin lisää tietoa siihen, miten opit ja kokemus vaikuttavat ketterän Scrum - menetelmän käyttöön.

Lähteet

Kirjalliset lähteet

Cohn, M. 2010. Succeeding with Agile. Software Development Using Scrum. Boston: Pearson Education.

Elssamadisy, A. 2009. Agile Adoption Patterns. A Roadmap to Organizational Success. Boston: Pearson Education.

Järvinen, P. & Järvinen, A. 2004. Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Kananen, J. 2008. Kvali - Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Yliopistopaino.

Ojasalo, K., Moilanen, T. & Ritalahti, J. 2009. Kehittämistyön menetelmät. Uudenlaista osaamista liiketoimintaan. Helsinki: WSOYpro.

Rico, D., Sayani, H. & Sone, S. 2009. The Business Value of Agile Software Methods. Maximizing RICO with Just-In-Time Process and Documentation. Fort Lauderdale: Ross Publishing.

Ruusuvuori, J., Nikander, P. & Hyvärinen, M. (toim.). 2010. Haastattelun analyysi. Tampere: Vastapaino.

Schwaber, K. & Beedle, M. 2002. Agile Software Development with Scrum. New Jersey: Prentice-Hall.

Yin, K. 2009. Case Study Research. Design and Methods. Fourth Edition. United States of America: SAGE Publications.

Sähköiset lähteet

Abrahamsson, P, Warsta, J, Sipponen, M & Ronkainen J. 2003. New Directions on Agile Methods: A Comparative Analysis. Viitattu 2.6.2011.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.7257&rep=rep1&type=pdf>

Ambler, S. 2006. An Introduction to Agile Modeling. Viitattu 2.6.2011.

<http://www.agilemodeling.com/essays/introductionToAM.htm>

Balijepally, V., Mahapatra, R. & Nerur, S. 2006. Assessing Personality Profiles of Software Developers in Agile Development Teams. Viitattu 4.6.2011.

[http://aisel.aisnet.org/cgi/viewcontent.cgi?article=3075&context=cais&sei-redir=1#search="agile+software+development+vs+traditional"](http://aisel.aisnet.org/cgi/viewcontent.cgi?article=3075&context=cais&sei-redir=1#search=)

Beck, K. 1999. Extreme Programming Explained. Viitattu 2.6.2011.

<http://www.mip.sdu.dk/~brianj/Extreme%20Programming%20Explained%20-%20Kent%20Beck%3B%20Addison-Wesley,%201999.pdf>

Benbasat, I., Goldstein, D. & Mead, M. 1987. The Case Research Strategy in Studies of Information Systems. Tulostettu 31.5.2011.

<http://wiki.bath.ac.uk/download/attachments/26150518/benbasat87-caseresearch.pdf>

Boehm, B. 1988. A Spiral Model Software Development and Enhancement. Tulostettu 31.5.2011.

<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/spiral.pdf>

Boehm, B & Hansen, W. 2001. The Spiral Model as a Tool for Evolutionary Acquisition. Viitattu 2.6.2011.

<http://nkhalid.seecs.nust.edu.pk/SE/software%20p.%20models%20readings/presentation%20.pdf>

Chau, T., Maurer, F. & Melnik, G. 2003. Knowledge sharing: Agile methods vs. Tayloristic methods. Tulostettu 15.2.2011.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.1144&rep=rep1&type=pdf>

Cho, J. 2008. Issues and Challenges of Agile Software Development with Scrum. Tulostettu 17.3.2011

http://www.iacis.org/iis/2008_iis/pdf/S2008_950.pdf

Cockburn, A. & Highsmith, J. 2001. Agile Software Development: The People Factor. Tulostettu 15.2.2011.

<http://www.uml.org.cn/softwareprocess/pdf/IEEEArticle2Final2.pdf>

Cohn, M. & Schwaber, K. 2003. The Need for Agile Project Management. Tulostettu 17.3.2011.

<http://www.mountangoatsoftware.com/system/article/file/14/ManagingAgileProjects.pdf?1267552461>

Darke, P., Shanks, G. & Broadbent, M. 1998. Successfully completing case study research: combining rigour, relevance and pragmatism. Tulostettu 31.5.2011.

http://www.uio.no/studier/emner/matnat/ifi/INF5500/h07/undervisningsmateriale/ISJ_case_study.pdf

Davis, A., Bersoff, E. & Comer, E. 1988. A Strategy for Comparing Alternative Software Development Life Cycle Models. Tulostettu 31.5.2011.

<http://users.tkk.fi/~tpunkka/strategy%20for%20comparing%20life%20cycle%20models.pdf>

Dube, L. & Pare, G. 2003. Rigor in Information Systems Positivist Case Research: Current Practices, Trends and Recommendations. Tulostettu 31.5.2011.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.108.2091&rep=rep1&type=pdf>

Highsmith, J. 2000. Using Adaptive Software Development to meet the challenges of a high-speed, high-change environment. Viitattu 2.6.2011.

<http://www.adaptivesd.com/articles/Dinosaurs.pdf>

Kruchten, P. 2011. What is Rational Unified Process? Viitattu 2.6.2011.

<http://perso.enstimac.fr/~journeau/GSI/MDP-info/Pr%20E9sentation%20RUP/WhatIsTheRationalUnifiedProcessJan01.pdf>

Mahnic, V. & Drnovscek, S. 2005. Agile Software Project Management with Scrum. Tulostettu 15.2.2011

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.901&rep=rep1&type=pdf>

Nerur, S., Mahapatra, R. & Mangalaraj, G. 2005. Challenges Of Migrating To Agile Methodologies. Tulostettu 26.2.2011.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.2745&rep=rep1&type=pdf>

Ness Technologies. 2010. Agile Software Best Practices: Agile and Globally Distributed Teams Can Coexist. Tulostettu 20.5.2011.

http://images.talentum.com/whitepaper/Ness_Technologies-June_2010_-_Agile.

Rational 1998. Rational Unified Process. Best Practices for Software Development Teams. Viitattu 2.6.2011.

http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

Samlinkin kotisivut. Infotietoa yrityksestä. Viitattu 9.5.2011.

www.samlink.fi

Samlinkin vuosikertomus. 2010. Viitattu 9.5.2011.

http://www.samlink.fi/linked/fi/samlink/Samlink-vuosikertomus_FIN_210311.pdf

Seidel, J. 1998. Qualitative Data Analysis. Tulostettu 31.5.2011.

<ftp://ftp.qualisresearch.com/pub/qda.pdf>

Sutherland, J. 2001. Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies. Tulostettu 15.2.2011.

http://codespeak.net/svn/pypy/tag/funding-final/agile_can_scale.pdf

Sutherland, J. 2003. The Roots of Scrum: How the Japanese experience changed global. Tulostettu 26.2.2011.

<http://www.gbcacm.org/sites/www.gbcacm.org/files/slides/5%20-%20Roots%20of%20Scrum.pdf>

Sureshchandra, K. & Shrinivasavadhani, J. 2008. Moving from Waterfall to Agile. Tulostettu 17.3.2011.

<http://140.115.155.41/agilemethod/download/2009papers/Moving%20from%20Waterfall%20to%20Agile.pdf>

The Agile Management Company. 2010. Agile Development: A Manager's Roadmap for Success. Tulostettu 20.5.2011.

http://images.talentum.com/whitepaper/Version_One-Agile_Managers_Roadmap.pdf

Turner, R. & Boehm, B. 2003. People Factor in Software Management: Lessons From Comparing Agile and Plan-Driven Methods. Tulostettu 20.4.2011.

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.952&rep=rep1&type=pdf>

Vinekar, V., Slinkman, C. & Sridhar, N. 2006. Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View. Tulostettu 17.3.2011.

<http://www.uta.edu/faculty/vinekar/Vinekar%20et%20al%202006.pdf>

Waterfall Model. 2011. Viitattu 31.5.2011.

<http://www.waterfall-model.com/the-waterfall-model-advantages-and-disadvantages/>

Julkaisemattomat lähteet

Asiakaskysely. 2011. Sähköisen työpöydän pilottiprojektissa mukana olleet asiakkaat 6.4 - 14.4.2011.

Asiakastytyväisyystutkimus. 2010. Oy Samlink Ab. Espoo.

Haastattelututkimus. 2011. Sähköisen työpöydän pilottiprojektissa mukana olleet asiantuntijat 11.4-19.4.2011. Oy Samlink Ab. Espoo.

Simola, P. 2011. Sovelluspalvelut yksikön johtajan haastattelu 11.3.2011. Oy Samlink Ab. Espoo.

Kuvat

Kuva 1. Scrum prosessi (Schwaber & Beedle 2002, 8.).....	18
Kuva 2. Scrumin runko (Mahnic & Drnovscek 2005, 2.)	19
Kuva 3. Tapaustutkimuksen vaiheet Yinin (2009) mukaan.....	32
Kuva 4. Tiedon analysointi prosessi (Seidel 1998, 2.)	38

Taulukot

Taulukko 1. Käsitteet ja niiden selitykset	10
Taulukko 2. perinteinen vastaan ketterä sovelluskehitys (Nerur ym. 2005, 75.).....	16

Liitteet

Asiantuntijahaastattelujen aihealueet ja alustavat kysymykset

Yleistä Scrumista tai projektin Scrum käytännöistä

- Mitä hyvää? Mitä etuja? Mitä huonoa?
- Kokeeko helpommaksi/paremmaksi vai huonommaksi tehdä kehitystä siten, että toteutetaan vähän kerrallaan ja nähdään tulos tietyn väliajoin, miksi?

Scrum vastaan vesiputousmalliin perustuva sovelluskehitysmenetelmä

- Mitä hyvää? Mitä huonoa? Mitä etuja?

Scrum tiimi

- Onko pitänyt tiimi työskentelystä, miksi?
- Miten tiimityöskentely on sujunut?
- Tehdäänkö työtä Scrumin edellyttämällä tavalla?
- Arvioiko tiimi työmäärät Sprinttiin?
- Onko kesken Sprintin lisätty työtä tai otettu sitä pois?
- Onko tiimillä ollut valtaa tehdä päätöksiä ja poistaa esteitä?
- Miten yhteistyö Jyväskylän tiimin kanssa on sujunut?

Daily Scrum

- Miten on sujunut?
- Onko keskitytty Scrumin mukaisiin kolmeen kysymykseen?

Scrum Master

- Onko auttanut ja tukenut tiimiä ja poistanut tarvittaessa esteitä?
- Onko vetänyt Daily Scrumit Scrumin sääntöjen mukaan?

Sprintti

- Onko tiimi saanut työskennellä rauhassa Sprintin aikana?
- Onko pysytty sovituissa tavoitteissa siitä mitä toteutetaan Sprintin aikana?

Product Owner

- Miten yhteistyö on sujunut?
- Onko selvittänyt tarpeellisia asioita?

Product Backlog

- Onko priorisoitu korkeimman prioriteetin mukaan?

- Onko Product Owner vastuussa ja ylläpitää?

Sprint Backlog

- Onko ylläpidetty?
- Onko tehtävät pilkottu ja arvioitu?

Sprint Review/Demotilaisuus

- Mitä hyvää? Mitä huonoa?
- Miten yhteistyö asiakkaiden kanssa on sujunut?

Sprint Review/Retrospective

- Mitä hyvää? Mitä huonoa?
- Mitä palaverissa tehtiin?
- Mietittiinkö Scrumin mukaisesti mikä meni edellisessä Sprintissä hyvin, mikä huonosti ja miten työskentelyä ja prosesseja voitaisiin kehittää?

Sprint Planning

- Mitä hyvää? Mitä huonoa?
- Mitä palaverissa tehtiin?

Organisaatio

- Tukeeko organisaatio ketterää toimintaa?

Lopuksi

- Mitä pitäisi jatkossa kehittää ja/tai parantaa tai erityisesti ottaa huomioon?
- Teetkö töitä mieluummin Scrum mallin kuin Samlinkin aikaisemman vesiputousmalliin perustuvan systeemityömenetelmän mukaisesti?