

## **Go Agile**

### **Projektimallin muutos ketterään Scrumiin**

Mervi Jäntti

Opinnäytetyö  
Toukokuu 2019  
Liiketalouden ala  
Tradenomi (AMK), Liiketalous

Tekijä(t) Jäntti, Mervi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Toukokuu 2019
	Sivumäärä 79	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi <b>Opinnäytetyön nimi: Go Agile</b> Mahdollinen alanimi: Projektimallin muutos ketterään Scrumiin		
Tutkinto-ohjelma Liiketalous		
Työn ohjaaja(t) Riku Ojanperä		
Toimeksiantaja(t) Yritys X		
<p>Tiivistelmä</p> <p>Kohdeyrityksen ohjelmistokehitysprojektissa on käyty muutosprosessia läpi, kun ohjelmiston tuottaja on siirtynyt vesiputousmallista ketterien menetelmien pariin. Tutkimuksen tarkoituksena oli tarkastella perinteistä vesiputousmallia ja ketterän kehityksen projektimallia Scrumia projektinhallintamenetelmänä sekä perehtyä ketterän kehityksen periaatteisiin. Pääpaino tutkimuksessa on Scrum-menetelmä, jota on toteutettu projektissa neljän kuukauden ajan. Toimeksiantajalle laadittiin raportti, jossa analysoitiin projektimallien välisiä eroja sekä tuotiin esiin, mitä vaikutuksia uusi malli Scrum on tuonut läpinäkyvyyden, laadun, nopeuden ja itseohjautuvuuden suhteen projektille ohjelmistokehitysprojektin jäsenten näkökulmasta.</p> <p>Tutkimuksellinen osuus toteutettiin laadullisella eli kvalitatiivisella tutkimusmenetelmällä ja tutkimusstrategiaksi valittiin tapaustutkimus. Tutkimuksen aineisto kerättiin haastattelemalla neljää projektiryhmän jäsentä, jotka toimivat tiiviissä yhteistyössä ohjelmiston tuottajan kanssa ja heillä on kokemusta molemmista projektimalleista.</p> <p>Tutkimus osoitti, että muutos on ollut oikeansuuntainen ja Scrumiin siirtyminen on tuottanut merkittäviä hyötyjä. Tutkimus osoitti myös, että ketteriin menetelmiin siirtyminen tuotekehityksessä heijastuu myös muuhun organisaatioon, jonka toimintatavoissa voi tulla tarvetta muutokselle. Ketteryyttä tarvitaan myös muulta liiketoiminnalta. Tutkimuksessa on nostettu esiin kehitysideoita Scrumista, joita myös muut organisaatioissa toimivat ryhmät voivat ottaa käyttöön soveltaakseen ketteryyttä päivittäisessä työarjessaan.</p>		
Avainsanat (asiasanat)		
Projektihallinta, vesiputous, ketterät menetelmät, Agile, Scrum		
Muut tiedot (Salassa pidettävät liitteet)		

Author(s) Jäntti, Mervi	Type of publication Bachelor's thesis	Date May 2019 Language of publication: Finnish
	Number of pages 79	Permission for web publication: x
	Title of publication <b>Title</b> Go Agile - Transition of project model to agile method Scrum	
Degree programme Business Administration		
Supervisor(s) Ojanperä, Riku		
Assigned by Company X		
Abstract  <p>The software development project of the target company has undergone a process of change when the software producer has moved from a waterfall model to agile methods. The purpose of the thesis was to examine the traditional waterfall model and the agile development project model, Scrum as a project management method and to create an overview of the principles of agile development. The focus of the thesis is the Scrum method, which has been implemented for four months in the project. The report was prepared for the target company by analyzing the differences between the project models and highlighting the effects of the new model on transparency, quality, speed and self-orientation from the point of view of the members of the software development project.</p> <p>The research was carried out with a qualitative research method and a case study was chosen as the research strategy. The research material was collected by interviewing four project team members who work in close collaboration with the software producer and have experience with the both project models.</p> <p>The study showed that the change has been in the right direction and the transition to Scrum has produced significant benefits. The research also showed that shifting to agile methods in product development is reflected in other organization also whose ways of working may need change. Agile is also needed for other businesses. The research has highlighted the elements of Scrum, which other groups in the organization can deploy to apply agile in their daily work routines.</p>		
Keywords/tags (subjects) Project Management, Waterfall, Agile, Agile methods, Scrum		
Miscellaneous (Confidential information)		

## Sisältö

<b>1</b>	<b>Johdanto .....</b>	<b>3</b>
<b>2</b>	<b>Tutkimusasetelma .....</b>	<b>7</b>
2.1	Tutkimusongelma .....	7
2.2	Tutkimusmenetelmät .....	8
<b>3</b>	<b>Vesiputousmalli projektinhallintamenetelmänä .....</b>	<b>17</b>
3.1	Vesiputousmenetelmän hyödyt .....	20
3.2	Vesiputousmallin haitat.....	21
<b>4</b>	<b>Ketterä kehitys eli Agile .....</b>	<b>23</b>
4.1	Ketterän kehityksen periaatteet .....	23
4.2	Siirtyminen ketteryyteen.....	28
<b>5</b>	<b>Scrum projektinhallintamenetelmänä .....</b>	<b>30</b>
5.1	Scrumin roolit .....	32
5.2	Scrumin tapahtumat.....	37
<b>6</b>	<b>Tutkimustulokset.....</b>	<b>43</b>
6.1	Projektijäsenten kokemukset läpinäkyvyydestä vesiputous vs. Scrum ....	44
6.2	Projektijäsenten kokemukset tuotosten laadusta vesiputous vs. Scrum .	48
6.3	Projektijäsenten kokemukset läpimenoajasta vesiputous vs. Scrum .....	53
6.4	Projektijäsenten kokemukset itseohjautuvuudesta vesiputous vs. Scrum 55	
6.5	Scrumiin siirtymisen myötä kohdatut haasteet .....	57
<b>7</b>	<b>Johtopäätökset.....</b>	<b>58</b>
<b>8</b>	<b>Pohdinta.....</b>	<b>69</b>
	<b>Lähteet .....</b>	<b>75</b>
	<b>Liitteet .....</b>	<b>78</b>
	Ketterään kehityksen takana olevat ydinarvot ja periaatteet.....	78
	Teemahaastattelurunko: .....	79

**Kuviot**

Kuvio 1: Kotterin malli muutoksen läpiviennin vaiheista.....	6
Kuvio 2: Vesiputousmallin vaiheittainen prosessi.....	18
Kuvio 3: Scrum prosessi ja aktiviteetit .....	38
Kuvio 4: Esimerkki käyttäjätarinasta .....	39
Kuvio 5: Tutkimustulokset Vesiputous vs. Scrum .....	44
Kuvio 6: Laatuun vaikuttavat tekijät vesiputous vs. Scrum.....	48
Kuvio 7: Tutkimustyön toteuttaminen sprinteissä.....	67

# 1 Johdanto

Lähes kaikissa projekteissa tarvitaan menetelmiä projektin ja prosessin kuvaukseen ja projektin eteenpäin viemiseen. Suuremmissa projekteissa käytetään erilaisia mallintamisen menetelmiä projektinhallintaan. Projektin luonteesta riippuu, millainen menetelmä sille valitaan. Projektille valittu menetelmä, mikä on yleisesti hyvin tunnettu, toimii viitekehyksenä projektin eteenpäin viemiselle. Tyypillisesti ohjelmiston kehitysprojekteissa käytetään järjestelmän toimittajan projektinhallintamenettelyä, ellei asiakkaalla ole vahvaa omaa menettelyä. (Myllymäki, Hinkka, Hirvensalo & Hämäläinen 2015, 66.) Projektimenetelmällä tarkoitetaan joukkoa käytäntöjä, joilla pyritään saavuttamaan projektille asetettu tavoite (Berkun 2006, 31).

Projektinhallinta pitää sisällään erittäin vaativaa ja monimutkaista toimintaa.

- Suunnittelu - mitä vaaditaan, että saadaan valmista. Tulee priorisoida, aikatauluttaa, miettiä resursseja ja riippuvuuksia projektin sisällä.
- Organisointi - kuinka asiat saadaan tehtyä. Tämä tarkoittaa projektitiimin jäsenien ja resurssien valintaa sekä tilojen, yhteydenpidon, toimintojen ja kokoontumisien organisoimista.
- Toteutus - saada tapahtumaan. Suunniteltujen toimenpiteiden toteutus.
- Seuranta - pitää hanke käynnissä. On seurattava suunnitelman edistymistä, mihin liittyy ongelmien tunnistamista ja niiden ratkaisemista, on annettava palautetta ja hallittava koko hankkeen etenemistä resurssien, tulevien vaatimusten ja riskien osalta. (Jones 2007, 7.)

Tutkimuksen kohdeyritys on Suomessa toimiva yritys, joka on osa kansainvälistä konsernia. Kohdeyritys tarjoaa luotonhallintapalveluja yrityksille Suomen markkinoilla. Yhtiöllä on toimipisteitä 24 maassa ja yrityksen palveluksessa työskentelee yli 8000 luotonhallintapalvelujen ammattilaista. Suomessa palveluita tuotetaan kolmesta toimipisteestä, jotka työllistävät noin 420 henkilöä. Yhtiö tarjoaa palveluita yli 20 000 yritykselle Suomessa.

Tutkimus kohdentuu Keski-Suomen toimipisteeseen ja siellä käynnissä olevaan ohjelmistokehitysprojektiin. Keski-Suomen toimipisteessä tarjotaan lasku- ja reskontrapalveluja, jotka mukautuvat monilla eri toimialoilla toimivien yritysten tarpeisiin.

Vuonna 2015 on aloitettu kohdeyksikössä ohjelmistokehitysprojekti, koska käytössä ollut ohjelmisto, jolla myyntireskontrapalveluja tuotetaan, oli todettu vanhanaikaiseksi ja kankeaksi ja uudet toiminnot osittain mahdottomia toteutettavaksi.

Jotta lasku- ja reskontrapalvelu olisi kilpailukykyinen palvelu nykyisillä markkinoilla, tavoitteeksi projektille asetettiin vanhan ohjelmiston vaihtaminen uuteen teknisesti modernimpaan ratkaisuun siten, että uusi ohjelmisto soveltuu samoihin liiketoimintaprosesseihin kuin vanha järjestelmä sekä antaa pohjan tulevaisuuden kehitykselle uusien toimintojen käyttöönottamiseksi ja uusien liiketoiminnan tarpeiden tukemiseksi. Uutta lasku- ja reskontrapalvelun ohjelmistoa on alettu kehittämään ”tyhjältä” ja ohjelmistoa on systemaattisesti rakennettu vaihe vaiheelta ottaen huomioon työntekijöiden toiveet ja asiakkaiden tarpeet. Uuden ohjelmiston avulla halutaan lisätä automatisointia mm. raportointiin ja suoritusten käsittelyyn sekä kasvattaa ohjelmiston käyttäjäystävällisyyttä.

Uuden kehitettävän ohjelmiston ympärille on perustettu vuonna 2015 Keski-Suomen toimipisteessä projektiryhmä, jonka jäsenistö koostuu kohdeyrityksen omassa palveluksessa olevista työntekijöistä. Projektiryhmän tehtäviin kuuluu määrityksien kirjoittaminen liiketoiminnan, asiakkaiden ja loppukäyttäjien tarpeiden perusteella kehitettävään ohjelmistoon sekä he tekevät loppukäyttäjätestausta kehitetyistä ominaisuuksista ennen tuotosten julkaisua ja käyttöönottoa. Projektiryhmä tekee tiivistä yhteistyötä ohjelmiston toimittajan kanssa.

Uutta ohjelmistoa on rakennettu yhteistyössä projektiryhmän ja järjestelmän tuottajan kanssa siten, että projektiryhmä määrittelee liiketoiminnan ja asiakkaan tarpeet, mitä ohjelmiston tulee tukea ja ohjelmiston tuottaja suunnittelee, kehittää ja toteuttaa arkkitehtuuriset ja teknologiset ratkaisut ohjelmistoon. Ohjelmiston toimittaja kehittää siis kohdeyksikön liiketoiminnan kannalta keskeistä ohjelmistoa.

Kohdeyrityksen ohjelmistokehitysprojektissa ohjelmiston tuottajan toteuttamasta vesiputousmallista saatujen vuosien kokemusten perusteella oli liiketoiminnan puolella todettu, että vesiputousmainen vaiheittainen prosessimalli ei enää palvelut liiketoiminnan ja asiakkaiden tarpeita. Ohjelmistokehitysprojekti kokonaisuudessaan käy laajaa muutosprosessia läpi, kun perinteisestä vesiputousmenetelmästä on siirrytty järjestelmän tuottajan taholla ketteriin menetelmiin, jolla on suoranainen vaikutus myös liiketoimintaan, asiakkaisiin ja muihin sidosryhmiin, joita projektiryhmä edustaa.

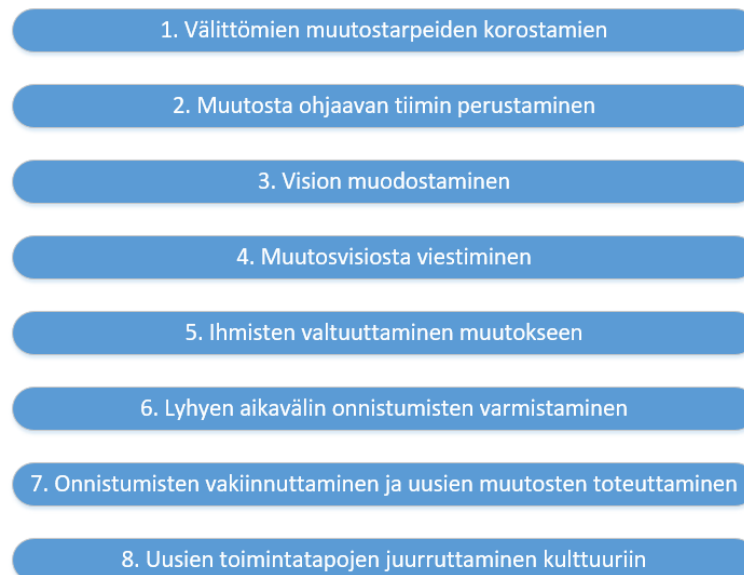
Tutkimuksen tavoitteena on tutkia kahta projektinhallintamenetelmää perinteistä vesiputousmallia ja ketterää Scrumia ja tuoda esiin vaikutuksia, mitä ohjelmiston tuottajan taholla tapahtunut projektinhallinnan muutos on tuonut ohjelmistokehitysprojektille projektijäsenten näkökulmasta. Tutkimus ei kohdistu vain Scrum-menetelmän käyttöönottoon vaan haastateltavia pyydettiin muistelemaan aikaisempaa projektinhallintamenetelmää vesiputousmallia ja tämän avulla tutkimuksessa toteutettiin kokemuspohjaista vertailua projektimallien välillä. Keskeistä tutkimuksessa on osoittaa, mitä hyötyjä projektinmallin muutos Scrumiin on tuonut liiketoiminnalle, kun järjestelmän toimittaja on siirtynyt toteuttamaan uutta menetelmää. Hyödyillä tarkoitetaan lisäarvoa, mitä Scrum-menetelmään siirtyminen on tuonut ohjelmistokehitysprojektiin, joka edustaa liiketoimintaa. Liiketoiminta, joka maksaa järjestelmän kehityksen kustannukset, on kuitenkin viime kädessä taho, joka päättää onko projekti onnistunut vai epäonnistunut (Myllymäki ym. 2015, 7).

Tutkimuksessa korostuu projektin jäsenten oma kokemus projektinhallintamenetelmistä ja uuden mallin tuomista vaikutuksista. Vaikka haastateltavien määrä on rajallinen, heillä on laaja kokemus, koska jokainen haastateltava on työskennellyt kohdeyrityksessä vuosia ja he ovat toimineet tutkimuksen kohteena olevassa projektissa pitkään ja täten osaavat tuoda esiin tutkimusilmiön kannalta oleellisia asioita. Tarkoituksena on luoda käsitys Scrumin tuomista hyödyistä ja haasteista kirjallisuuskatsauksen ja projektiryhmässä toteutettavan tapaustutkimuksen tutkimustulosten pohjalta. Tutkimuksen on tarkoitus myös osoittaa, voidaanko ketterästä projektimallista löytää koko organisaatiota hyödyntäviä käytäntöjä, jotta ketteryyttä voitaisiin lisätä koko organisaatiossa.



Odotettuja hyötyjä ketterissä menetelmissä ovat tehokkuus, laatu, läpinäkyvyys ja tiimin itseohjautuvuuden parantuminen ja nämä ovat myös esiolettamuksia tutkimukselle. Tarkoituksena tutkimuksen lopputuloksena esittää konkreettisia projektimallin muutoksen tuomia vaikutuksia, hyötyjä ja mahdollisia haittoja juurikin näistä näkökulmista. Tutkimuksesta rajattiin pois kustannusten arviointi, joten tutkimuksen ulkopuolelle on jäänyt taloudellisen hyödyn arviointi.

Suunniteltu muutos ei toteudu itsestään ja kaikkiin merkittäviin muutoksiin liittyy muutosvastarintaa ja tämän vuoksi muutoksen läpivienti edellyttää työtä, mikä on hyvin usein käytännönläheistä työtä sen sijaan, että esitetään asioita teoriassa. John P. Kotter esitteli vuonna 1995 työkalun, jossa tarvittavat työvaiheet on jäsennetty muutoksen läpiviemiseen. Kotterin mallia on käytetty johtavana muutosjohtamisen teoriana jo vuosia. Kotterin malli sisältää kahdeksan eri vaihetta (kuvio 1).



Kuvio 1: Kotterin malli muutoksen läpiviennin vaiheista

Lähtökohtana tutkimusraportille toimii myös Kotterin muutosjohtamisen mallin mukaisen muutoksen läpiviemisen kuudes vaihe. Lyhyen aikavälin onnistumisia tulee osoittaa, jotta voidaan nähdä, että muutoksen eteen tehty tuo on kannattavaa (Kotter 2012, 23). Tarkoituksena tutkimuksessa on tuoda esiin toimeksiantajalle lyhyen aikavälin näyttöjä ja onnistumisia muutoksesta, joka antaa johdolle kuvaa siitä, onko muutoksella saatu toivottuja hyötyjä ja tuoda lyhyen ajan onnistumisia näkyville vai tuleeko visiota muuttaa jollain tavalla.

## 2 Tutkimusasetelma

Tämä luku sisältää kuvauksen lähtökohtana tutkimukselle toimivasta tutkimusongelmasta ja työkaluista, joita on käytetty tutkimuksessa ongelman ratkaisemiseksi. Tutkimusongelmasta on johdettu tutkimuskysymykset, joiden avulla tutkimusongelma on ratkaistu ja kerrotaan perustelut tutkimusmenetelmien valinnalle.

### 2.1 Tutkimusongelma

Edellytys opinnäytetyön onnistumiselle on tutkimusongelma ja sen määrittely. Tutkimusongelma kiteyttää koko tutkimuksen tarkoituksen ja tavoitteet, joten tutkimusaihe kiteytetään tutkimusongelmaksi. Tutkimusongelma on kuin yrityksen missio - kaikki tulee kiteyttää yhteen lauseeseen. (Kananen 2008, 51.) Tutkimusaihetta rajataan tarkentamalla ajatusta siitä, mitä halutaan tietää tai mitä halutaan osoittaa kerätyllä aineistolla (Hirsjärvi, Remes & Sajavaara, 81). Tämän opinnäytetyön taustalla oleva tutkimusongelma on: mitä hyötyjä ja haasteita projektimenetelmän muutos vesiputousmallista Scrumiin on tuonut ohjelmistonkehitysprojektille?

Jotta tutkimusongelmaan saadaan ratkaisu, tulee ongelma muuttaa tutkimuskysymyksi. Tutkimuskysymykset ovat tutkijan työkaluja, jotka tarvitsevat lisäksi tarkennettuja kysymyksiä, joilla tuotetaan itse aineisto ja jonka avulla saadaan vastaus tutkimusongelmaan. (Kananen 2015, 55, 57.) Tutkimusongelmalle on asetettu seuraavat kolme tutkimuskysymystä:

1. Millaisia kokemuksia projektin jäsenillä on vesiputousmallin mukaisesta projektinhallinnasta?
2. Millaisia vaikutuksia projektimallin muutos Scrumiin on projektille tuonut?
3. Mitä haasteita projektimallin muutoksen jälkeen on kohdattu?

Tutkimuksen yhteydessä toteutetaan kirjallisuuskatsaus, mikä käsittelee aiheeseen liittyvää teoriaa. Teoriaosuuden avulla luodaan käsitys molemmista projektinhallintamenetelmistä. Teoriaosuus etenee siten, että ensiksi käsitellään vesiputousmallia projektinhallintamenetelmänä, jonka jälkeen käsitellään Agilen eli ketterän kehityksen periaatteita sekä organisaation siirtymistä ketteryyteen. Ketterää kehitystä haluttiin käsitellä omana osanaan, koska Scrum on yksi ketterän kehityksen metodeista. Päähuomio kohdistuu kirjallisuuskatsauksessa Scrum-menetelmään, mikä on kirjallisuuskatsauksen viimeinen osa.

## 2.2 Tutkimusmenetelmät

### **Tutkimusote**

Laadullisen eli kvalitatiivisen tutkimuksen tavoitteena on tutkittavan ilmiön kuvaaminen, ymmärtäminen ja tulkinnan antaminen, joten laadullinen menetelmä pyrkii ilmiön syvälliseen ymmärtämiseen. Laadullinen tutkimus tutkii yksittäistä tapausta kuin taas määrällinen eli kvantitatiivinen tutkimus tutkii tapausten joukkoa. Laadullisessa tutkimuksessa käytetään sanoja ja lauseita, kun määrällinen tutkimus pohjautuu lukuihin. Näin ollen laadullisen tutkimuksen tuloksia ei voida yleistää siten kuin suuria joukkoja tutkivassa määrällisessä tutkimuksessa. (Kananen 2014, 19–20, 24.)

Laadullinen ja määrällinen tutkimus eivät eroa rakenteeltaan tai tutkimusprosessin osalta toisistaan, mutta ero tutkimusotteiden välillä tulee tutkimuksen lähtökohdista. Määrällinen tutkimus edellyttää teoriaa ja esiymmärrystä tutkittavasta ilmiöstä, jotta niitä voidaan ruveta mittaamaan määrällisin tutkimuksen keinoin kuten kyselylomakkeella, mikä mahdollistaa lukujen ja niiden välisten suhteiden käsittelyn. Olemassa

oleva teoria ohjaa määrällisen kysymysten asettelua. (Kananen 2015, 198.) Laadullinen tutkimus muodostaa toisen ääripään, jossa tutkimusilmiöstä ei ole paljoa tietoa (Kananen 2008, 68).

Tätä tutkimusta voi luonnehtia laadulliseksi tapaustutkimukseksi, sillä tavoitteena on tutkittavan ilmiön kuvaaminen, syvällinen ymmärtäminen ja tulkinnan antaminen ja tarkastelun kohteena on yksi kohdeyrityksen ohjelmistokehitysprojekti. Tapaustutkimuksen tarkoitus on antaa yksityiskohtaista tietoa yksittäisestä tapauksesta tai pienestä joukosta toisiinsa suhteessa olevia tapauksia (Hirsjärvi ym.1997, 134). Tapaustutkimus on luonteeltaan kuvailevaa ja tarkoituksena onkin kuvailla projektimallien eroavaisuuksia ja uuden projektimalliin liittyviä hyötyjä ja haasteita projektin jäsenten näkökulmasta yhdessä yksittäisessä ohjelmistokehitysprojektissa.

Kvalitatiivinen tutkimusote valittiin sen vuoksi, koska tässä tutkimuksessa tutkitaan prosesseja, ollaan kiinnostuneita siitä, kuinka ihmiset kokevat ja näkevät asioita. Määrällisen tutkimuksen keinoin, tilastollisin menetelmin tämän kaltaisiin asioihin on lähes mahdotonta pureutua, koska määrällisessä tutkimuksessa työvälineinä käytetään mm. kyselyitä. (Kananen 2008, 24.) Kananen toteaa vielä, että laadullisessa tutkimuksessa tutkija on suorassa kontaktissa tutkittavan kanssa, joka toteutetaan menemällä ilmiön pariin ”kentälle” haastattelemaan. Tarkoituksena laadullisessa tutkimuksessa on saada irti yhdestä havaintoyksiköstä mahdollisimman paljon. (Kananen 2014, 19–20.)

### **Aineistonkeruumenetelmät**

Aineistonkeruumenetelmillä kerätään tutkimusaineisto ja tutkimusote määrittää aineistonkeruumenetelmän. Laadullisessa tutkimuksessa käytettäviä aineistonkeruumenetelmiä ovat:

- dokumentit
- havainnointi
- haastattelu
- sähköpostihaastattelu. (Kananen 2015, 81.)

Teemahaastattelu on yleisin laadullisissa tutkimuksissa käytetyistä tiedonkeruumenetelmistä. Teemahaastattelun avulla tutkija pyrkii saamaan käsityksen tutkittavasta ilmiöstä, jossa on mukana ihminen ja hänen toimintansa. Teemahaastattelu valittiin tutkimuksen aineistonkeruumenetelmäksi, koska ilmiötä ei tunnettu hyvin ja haluttiin saada ilmiöstä mahdollisimman syvä ymmärrys. (Kananen 2014, 71–72, 76.) Tutkittavaa ilmiötä lähestyttiin eri näkökulmista eli teemoista, jolloin päästiin keskustelemaan ilmiön kanssa ja minkä avulla saadaan ymmärrystä vieraaseen aiheeseen (Kananen 2010, 53). Teemalla tarkoitetaan ilmiön osaa eli aihekokonaisuutta (Kananen 2017, 49). Teemahaastattelu tarjoaa tutkijalle riittävästi väljyyttä ja antavat tilaisuuksien käyttää rajaamismahdollisuuksia teemoittelulla. Teemojen tarkoitus on varmistaa, että kaikkia ilmiöön liittyviä osa-alueita sivutaan (Kananen 2009, 64–65.)

Hirsjärvi ja muut (1997, 208–209) ovat todenneet, että teemahaastattelua pidetään lomake- ja avoimen haastattelun välimuotona. Tyypillistä teemahaastattelulle on, että haastattelun aihepiirit eli teema-alueet ovat tiedossa, mutta kysymysten tarkka muoto ja järjestys puuttuvat. Aineistosta voidaan laskea frekvenssejä ja tuloksia voidaan analysoida ja tulkita monella tavalla.

Teemahaastattelu koostuu nimensä mukaisesti eri teemoista, joita haastatellaan. Haastattelu tulee toteuttaa kasvotusten molempien osapuolten fyysisesti läsnä ollessa (Kananen 2014, 76). Tutkijan tulee olla perehtynyt aiheeseen ja olla ennakkokäsitys ilmiöstä, jotta hän voi laatia teemat teemahaastattelulle. Teemahaastattelu etenee siten, että tutkija avaa keskustelun teemoittain ja tekee tarvittaessa tarkentavia kysymyksiä. Usein vastauksista syntyy uusia keskustelunaiheita ja keskustelu etenee luonnollista polkua. Etukäteen suunnitellut teemat auttavat siinä, että kaikista osa-alueista tulee keskusteltua, mutta keskustelu etenee kuitenkin haastateltavan ehdoilla. (Kananen 2017, 50.)

Yksi laadullisen tutkimuksen tyypillinen piirre on, että tutkimuksen kohdejoukko valitaan tarkoituksenmukaisesti, ei satunnaisotantaa käyttäen (Hirsjärvi ym. 1997, 164). Laadullisessa tutkimuksessa keskitytään myös varsin pieniin määriin tapauksia pyrkien analysoimaan niitä mahdollisimman kattavasti. Näin ollen aineiston tieteellisyy-

den kriteeri ei ole kohdejoukon määrä vaan laatu (Eskola & Suoranta 1998, 18.) Haastateltaviksi valitaan henkilöt, joita tutkimusilmiö koskettaa ja jotka tietävät asiasta parhaiten (Kananen 2015, 146.)

Tapaustutkimuksen kohteena toimii kohdeyrityksen neljä vuotta käynnissä ollut järjestelmän kehitysprojekti ja tutkimuksen havaintoyksiköiksi on valittu harkinnanvaraisesti henkilöt, jotka ovat toimineet projektissa useita vuosia. Tutkimuksessa haastateltiin neljää projektinjäsentä, joilla on laaja näkemys, tietämys ja ennen kaikkea kokemusta molemmista projektinhallintamenetelmistä sekä teoreettisesti että käytännön tasolla. Kaikki haastateltavat ovat toimineet projektissa pitkään ja ovat olleet muutosprosessissa mukana. Haastateltavat toimivat liiketoiminnan edustajina projektissa ja heillä on kokemusta vesiputousmallin mukaisesta ohjelmistokehityksestä ja näkemystä niistä vaikutuksista, mitä ohjelmiston toimittajan siirtyminen Scrum-menetelmään on projektiin heijastanut ja ovatko Scrumin arvolupaukset toteutuneet heidän näkökulmastaan. Tutkimuksessa on haastateltu yhtä henkilöä lukuun ottamatta kaikki henkilöt, jotka on voitu tutkimusongelman ratkaisemiseksi haastatella.

Haastattelut toteutettiin huhtikuussa 2019 kahden viikon sisällä. Ennen haastatteluja haastateltavilta on kysytty lupa haastattelulle ja kaikki haastateltavat olivat kiinnostuneita osallistumaan. Tämän jälkeen haastateltaviin oltiin yhteydessä sähköpostilla, jossa kerrottiin tarkemmin tutkimuksen taustasta ja tarkoituksesta. Saateviestissä tuotiin ilmi käytännön järjestelyt, kuten haastattelun ajankohta, paikka ja arvioitu kesto sekä haastateltaville annettiin mahdollisuus osallistua haastatteluun myös Skype-palaverin välityksellä. Viestissä korostettiin haastattelun luottamuksellisuutta ja anonyymiyttä eli sitä, ettei haastateltavien henkilöllisyyttä ja tutkimuskohteena olevaa yrityksen nimeä ja projektia tuoda esiin tutkimustiedoissa eikä tutkimustuloksissa.

Haastattelujen luotettavuutta pyrittiin vahvistamaan suunnittelemalla haastattelut huolellisesti. Haastatteluiden teemat ja niistä johdetut haastattelukysymykset muodostettiin aikaisemmin syvällisesti perehdytyn tutkimusilmiön teoriatietoon pohjautuen. Ennen varsinaisia haastatteluja haastattelukysymykset testattiin ja muokattiin koehaastattelun avulla. Koehaastattelu toteutettiin erään IT-konsultin kanssa, jolla oli

kokemusta molemmista projektinhallintamenetelmistä usean vuoden ajalta. Hän on toiminut usean eri yrityksen erilaisissa IT-projekteissa, mutta hän itse ei varsinaisesti toiminut tämän tutkimuksen kohteena olevassa projektissa.

Haastattelut toteutettiin kunkin haastateltavan kanssa yksilöhaastatteluna, koska tutkimuksessa toivottiin jokaisen henkilökohtaista näkemystä sekä vesiputousmallin aikaisista kokemuksista että uuden Scrum-mallin tuomista vaikutuksista. Kahden kanssa haastattelu tapahtui kohdeyrityksen neuvottelutiloissa ja kahden haastateltavan kanssa Skype-palaverin välityksellä. Haastattelut kestivät keskimäärin 45 minuuttia per haastattelu. Haastattelujen alussa haastateltaville kerrottiin vielä tutkimuksen tarkoituksesta, luottamuksellisuudesta ja haastattelun kulusta. Haastattelussa käytettiin kaikkien haastateltavien kanssa samaa teemahaastattelurunkoa, mikä eteni siten, että ensimmäisenä kysymyksenä kartoitettiin haastateltavan taustaa ja loput kysymykset liittyivät tutkittavan ilmiön keskeisiin teemoihin.

Haastattelujen avulla pyrittiin saamaan esiin kokemusperäistä tietoa projektinhallintamenetelmistä ja niiden toimivuudesta. Etukäteen valitut teemat perustuivat tutkimuksen viitekehykseen eli tutkittavasta ilmiöstä jo tiedettyyn. Teemahaastattelu eteni tiettyjen etukäteen valittujen teemojen ja niihin liittyvien tarkentavien kysymysten varassa. (Tuomi & Sarajärvi 2009, 75.) Lisäkysymyksillä pyrittiin saamaan lisävalaistusta pohdittavaan ilmiöön (Alasuutari 1994, 270). Teemahaastattelussa käytettiin suppilotekniikkaa, mikä tarkoittaa, että yksittäisen teeman puitteissa edettiin yleisestä yksityiskohtiin, minkä avulla täsmennettiin ymmärrystä ilmiöstä (Kananen 2015, 150).

Keskustelua käytiin teemoittain ja tehtiin tarvittaessa tarkentavia kysymyksiä. Usein vastauksista syntyi uusia keskustelunaiheita ja keskustelu eteni sujuvasti. Kaikki haastateltavat olivat avoimia ja rehellisiä. Tähän vaikutti seikka, että haastattelu tilanne pyrittiin saamaan mahdollisimman avoimeksi ja rennoksi. Haastateltavan avoimuuteen vaikutti myös se, että haastattelija oli tuttu, jolle pystyi vapaasti kertomaan kokemuksistaan. Yhteenkään kysymykseen ei kieltäydytty vastaamatta. Osaltaan haastateltavien avoimuuteen vaikutti myös tieto siitä, että tutkimustiedot ja -tulokset käsitellään haastateltavien osalta nimettöminä.

Haastattelun nauhoittamisessa käytettiin tallennusvälinettä, minkä toimivuus varmistettiin muutaman kerran haastattelutilanteessa. Tallenteet muutettiin tekstimuotoon litteroimalla sanatarkasti heti haastattelupäivänä tai sitä seuraavana päivänä, mikä osaltaan paransi haastattelun laatua ja mikä inspiroi alustavan analyysin tekemistä jo tässä vaiheessa sekä työstämään aineistoa myöhempää analyysiä varten. Tallennus mahdollisti myös tietojen tarkastelun jälkikäteen.

### **Analyysimenetelmät**

Tässä tutkimuksessa analyysi on tehty aineistolähtöisesti eli sen mukaan, mitä aineistosta on noussut esiin (Kananen 2014, 103–104, 107–108). Kaikki, mitä ilmiönä tai aikaisempien tutkimusten perusteella tiedetään, on pyritty rajaamaan analyysin ulkopuolelle, niin ettei se vaikuttaisi analyysiin. Aineistolähtöisessä analyysissä pyritään ymmärtämään aineiston perusteella mitä tutkittavat asiat merkitsevät tutkittavalle. Tutkittavia tulee ymmärtää heidän omasta näkökulmasta analyysin kaikissa vaiheissa. (Tuomi & Sarajärvi 2009, 96, 113.) Teorialähtöinen analyysi poissuljettiin tämän tutkimustulosten analyysimenetelmänä, koska menetelmä nojaa tiettyyn teoriaan ja aineiston analyysiä ohjaa valmis aikaisemman tiedon perusteella luotu kehys. Teorialähtöisessä analyysissä päättelyn logiikka linkittyy deduktiiviseen päättelyyn, mihin kuuluu ajatus yleisestä yksityiseen ja palautus sellaisenaan yksityisestä yleiseen (Tuomi & Sarajärvi 2009, 98).

Usein laadullisessa tutkimuksessa, missä aineistoa on kerätty havainnoimalla, haastatteleamalla ja keräämällä aiheeseen liittyviä dokumentteja, aineistomäärät muodostuvat niin suuriksi, ettei pelkästään lukemalla, kuuntelemalla tai katselemalla päästä ratkaisuun (Kananen 2014, 101). Laadullisen aineiston analysointi edellyttää aineistojen yhteismitallistamista, mikä tarkoittaa sitä, että tallennetut aineistot muutetaan tekstimuotoon, jolloin niitä voidaan käsitellä erilaisilla analysointimenetelmillä (Kananen 2015, 83). Tallennettu laadullinen aineisto on tarkoituksenmukaista kirjoittaa puhtaaksi sanatarkasti litteroimalla. Tutkimustyötä tehdessä on analyysiin ryhdyttävä heti, jotta aineisto inspiroi vielä tutkijaa ja aineistoa voidaan tarvittaessa vielä täydentää tai selventää (Hirsjärvi ym.1997, 224).



Litteroinnissa voidaan käyttää eri tekniikoita ja tasoja. Sanatarkka litterointi merkitsee tarkinta tasoa, jossa kirjataan ylös myös äänenpainot, eleet ja äännähdykset. Tässä tutkimuksessa haastatteluista saatu aineisto on litteroitu käyttäen yleiskielistä litterointia, missä teksti on muutettu kirjakielelle poistaen murre- ja puhekielen ilmaiset (Kananen 2014, 102.) Tutkimusaineistoa litteroitaessa kaikki erilaiset äännähdykset on jätetty pois, koska niillä ei ole merkitystä viestin välittymisen kannalta ja poistamisen myötä aineisto tekstisisällön osalta saatiin sujuvammaksi ja helppolukuisemmaksi. Aineistoa läpikäydessä eroteltiin asiat, jotka sisältyivät kiinnostuksen kohteiksi. Tämä tehtiin keräämällä merkkamalla asiat yhteen ja erottamalla kiinnostavat asiat muusta aineistosta (Alasuutari 2009, 92). Koska kyse oli laajemmasta aineistosta, käytettiin tekstin käsittelyssä tekstinkäsittely- ja taulukkolaskentaohjelmia (Kananen 2015, 83).

Analyysimenetelmiä on monia ja tarkoituksena oli valita sellainen analyysitapa, mikä parhaiten tuo vastauksen tutkimusongelmaan. Aineisto luettiin useaan kertaan läpi ja tutustuttiin aineistoon teemoitellen ja tehtiin sitä mukaan jo alustavia valintoja analyysitavasta. Teemoittelussa painottuu, mitä kustakin teemasta on sanottu. (Hirsjärvi ym. 1997, 224.)

Litteroinnin ja aineistoon perehtymisen jälkeen teksti tiivistettiin koodaamalla, jonka myötä aineisto saatiin ymmärrettävään ja käsiteltävään muotoon. Tässä tutkimuksessa haastateltavien vastaukset tallennettiin samaan tekstinkäsittelyohjelmaan, jonka jälkeen samaan teemaan kuuluvat neljän eri haastateltavan vastaukset siirrettiin peräkkäin laskentataulukoon. Tämän mahdollisti vastausten vertailun keskenään sekä yhteneväisyyksien ja eroavaisuuksien etsimisen.

Samaa tarkoittavat asiat yhdistettiin samalla koodilla ja myös sellaiset asiat, joilla oli yhteinen tekijä. Koodauksen avulla aineisto saatiin tiivistettyä ja yksinkertaistettua. Koodauksen jälkeen aineisto analysoitiin luokittelemalla, mikä tarkoittaa joko samaa tarkoittavien käsitteiden tai eri käsitteiden yhdistämistä yhden käsitteen alle, jolloin niistä sai muodostettua loogisen kokonaisuuden (Kananen 2014, 113.) Samaa tarkoittavat käsitteet ryhmiteltiin luokaksi ja nimettiin luokan sisältö kuvaavalla nimellä. Luokkien nimeksi tämän tutkimusaineiston analyysissä muodostui: Läpinäkyvyys,

laatu, läpimenoaika, itseohjautuvuus ja haasteet. Luokittelun avulla luotiin pohja tutkimuksen perusrakenteelle sekä avustavia kuvauksia tutkittavasta ilmiöstä (Tuomi & Sarajärvi 2009, 110.)

### **Tutkimuksen luotettavuus**

Validiteetti tarkoittaa, että tutkimuksessa on tutkittu sitä mitä on luvattu ja reliabiliteetti tutkimuksen toistettavuutta. Käsitteet ovat syntyneet määrällisen tutkimuksen piirissä ja ne vastaavat määrällisen tutkimuksen tarpeita (Tuomi & Sarajärvi 2009, 136.) Myös Kananen 2014 mukaan em. luotettavuusmittarit on omaksuttu määrälliseen tutkimuksen luonnontieteistä, jossa tutkittavat ilmiöt voidaan viedä laboratorioolosuhteisiin, jolloin voidaan tehdä tarkkoja kokeita ja mittauksia, jotka ovat uusittavissa eivätkä tutkimustulokset muutu. Sen sijaan, kun tutkitaan ihmisen käyttäytymistä, ajattelua ja kokemuksia, ei tutkittavaa ilmiötä voida viedä laboratorioolosuhteisiin, koska ihmisten toiminta ei ole kaavamaisista. Objektiivisen luotettavuuden tavoittaminen laadullisessa tutkimuksessa onkin lähes mahdotonta ja luotettavuus onkin tutkijan arvioinnin ja näytön varassa. Laadullisessa tutkimuksessa luotettavuuskysymykset on otettava huomioon jo suunnitteluvaiheessa. (Kananen 2014, 144–146.)

Kananen (2014, 122) kuitenkin huomauttaa, että validiteetti voidaan jakaa myös alalajeihin ja tällöin voidaan puhua ulkoisesta ja sisäisestä validiteetista. Ulkoinen validiteetti tarkoittaa tulosten yleistettävyyttä ja sisäinen validiteetti tulkinnan virheettömyyttä. Tämän tutkimuksen tarkoituksena oli tutkia, mitä hyötyjä ja haasteita projektimenetelmän muutos vesiputousmallista Scrumiin on tuonut ohjelmistonkehitysprojektille projektinjäsenten näkökulmasta sekä tarkastella miten projektimallien vaiheet ja toimintatavat poikkeavat toisistaan vertailun tekemiseksi.

Kananen (2014, 151) mukaan laadullisessa tutkimuksessa käytettäviä luotettavuuskriteereitä ovat

- vahvistettavuus
- arvioitavuus/dokumentaatio
- tulkinnan ristiriidattomuus

- luotettavuus tutkitun kannalta
- saturaatio eli kylläntyminen.

Vahvistettavuus voidaan toteuttaa luetuttamalla tutkimusaineisto ja siitä tehdyt tulokset henkilöillä, joita on haastateltu. Haastateltavan luettua ja vahvistamalla tekstiä ja tulkinnan tukee tutkimuksen luotettavuutta tutkitun kannalta. Kun tutkittava vahvistaa tutkijan kuvauksen ja tulkinnan oikeaksi, tilanne on tutkijan kannalta ongelmaton. (Kananen 2014, 151.)

Arvioitavuudella tarkoitetaan sitä, että perehdytään riittävän tarkasti tutkimusmenetelmiin, jotta ollaan selvillä menetelmien soveltuvuudesta erilaisiin tutkimusilmiöihin. Tutkimuksen eri vaiheissa tulee perustella tehdyt ratkaisut ja valinnat. Riittävä dokumentaatio tuo tutkimukselle uskottavuutta. Tutkimuksen alkuvaiheesta lähtien päiväkirjaa pitämällä kaikesta toiminnasta, mikä liittyy opinnäytetyöhön, kannattaa pitää. (Kananen 2014 153.) Laadullisen tutkimuksen luotettavuutta parantaa tutkijan tarkka selostus tutkimuksen toteuttamisesta (Hirsjärvi ym. 1997, 232).

Laadullisessa tutkimuksessa on useista eri lähteistä kerättyä aineistoa, jolloin tulkinta tehdään monilähteisenä synteesisinä. Yksittäisen aineisto tulkinnassa voi esiintyä eroja erityisesti, jos aineisto on kerätty teemahaastattelulla. Samasta aineistosta voidaan tehdä eri tulkintoja riippuen tarkastelukulmasta ja tutkimusongelmasta. Tulkinnan ristiriidattomuudesta voidaan puhua silloin, kun toinen tutkija päätyy samaan lopputulokseen tutkijan kanssa. Kahden tutkijan päätyminen samoihin johtopäätöksiin lisää tutkimuksen luotettavuutta. (Kananen 2014, 153.)

Saturaatiolla eli kylläntymisellä voidaan vahvistaa laadullisen tutkimuksen luotettavuutta. Saturaatiolla tarkoitetaan sitä, että eri lähteiden tuottamat tutkimustulokset alkavat toistua. Uusia havaintoyksiköitä otetaan tutkittavaksi niin kauan, kuin ne tuovat jotakin uutta tutkimukseen. Kun vastaukset alkavat toistaa itseään, on saavutettu saturaatio. (Kananen 2014, 154.)

## Aiemmat tutkimukset

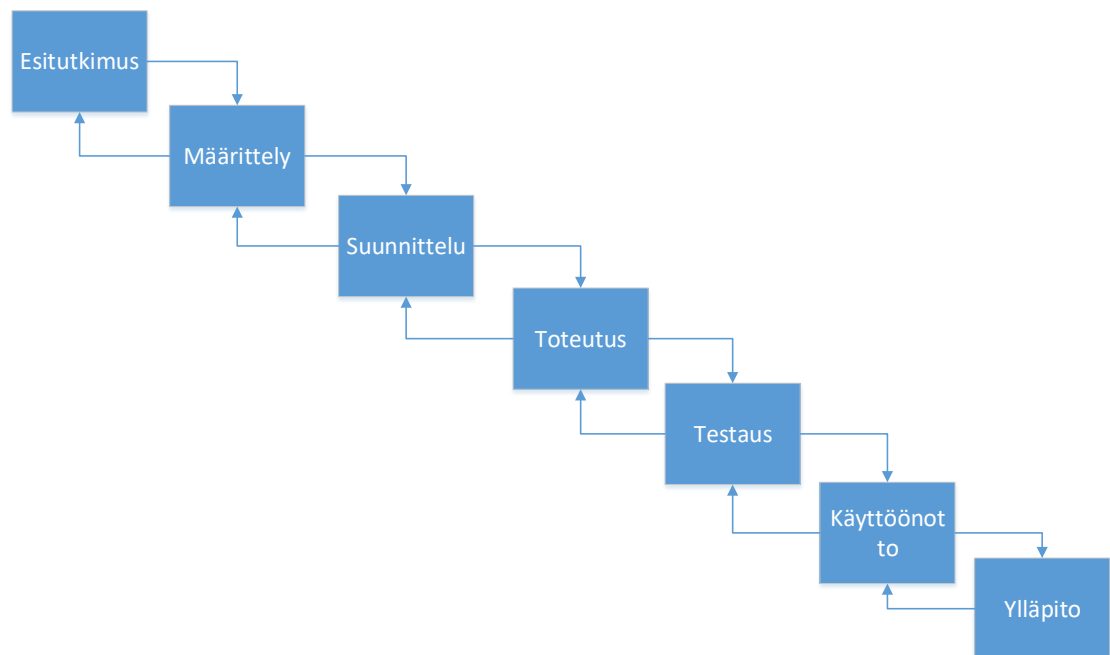
Vuori (2014) omassa opinnäytetyössään tutki projektinhallinnan muutosta perinteisestä vesiputousmallista ketterään Scrum-menetelmään. Tutkimuskohteena oli ohjelmistotalo Metsys Oy, joka on asiantuntijayritys teollisuuden tietojärjestelmistä. Tutkimuksessa seurattiin Scrum-menetelmän toimivuutta noin puoli vuotta kestävässä pilottiprojektissa ja tavoitteena oli selvittää projektin onnistumista ja soveltumista kohdeyrityksen toimintaympäristöön ja työtapoihin. Tutkimustuloksissa esitetään analyysi projektin onnistumisesta, tarkastellaan Scrum-menetelmän hyötyjä ja haasteita sekä pohditaan Scrum-menetelmän sopivuutta kohdeyrityksen toimintaympäristöön mm. laadun ja kustannusten näkökulmasta. Tutkija itse toimi projektissa Scrum Masterina. (Vuori 2014, 6, 54–62).

Ylemmässä opinnäytetyössään Pylvänäinen (2009) tutki millaisia muutoksia sidosryhmien toimintaan ketterien menetelmien käyttöönotto on tuonut ja ulottuivatko vaikutukset myös johtamiseen. Tutkimuskohteena oli liiketoimintayksikkö X, jossa ketteriä menetelmiä oli otettu käyttöön. Tapaustutkimuksessa selvitettiin minkälaisia vaikutuksia ja muutoksia kohdeyrityksen sidosryhmien toimintaan ketterien menetelmien käyttöönotto toi. (Pylvänäinen 2009, 3,7)

## 3 Vesiputousmalli projektinhallintamenetelmänä

Perinteisesti ohjelmistoprojektit on toteutettu 1970-luvulla kehitettyä ohjelmistotuotantoprosessia, jossa suunnittelu- ja toteutusprosessi etenee vaiheittain alaspäin. Projektimallia kutsutaan vesiputoukseksi ja sen mukaan projekti viedään lävitse lineaarisesti etukäteen määriteltyjen vaiheiden mukaisesti alaspäin kuten vesiputouksessa. Vesiputousmallista on useita versioita, mutta yleisesti malliin kuuluu ainakin määrittely, suunnittelu, toteutus, käyttöönotto ja ylläpito. (Haikala & Märijärvi 2004, 35.) Kuviossa 2 seuraavalla sivulla esitetään vesiputousmallin vaiheet. Kun yksi vaihe on saavutettu, voidaan siirtyä seuraavaan vaiheeseen. Vesiputousmalli edellyttää,

että yksi vaihe toteutetaan valmiiksi ennen kuin voidaan siirtyä seuraavaan vaiheeseen. Paluu edelliseen vaiheeseen on vielä mahdollista, mutta sitä aikaisempiin vaiheisiin todella haasteellista.



Kuvio 2: Vesiputousmallin vaiheittainen prosessi (Vuori 2014, muokattu)

Esitutkimusvaiheessa selvitetään yleiset, korkeamman tason vaatimukset, mikä tarkoittaa käytännössä sitä, kartoitetaan asiakkaan tarpeet, mutta ei vielä oteta kantaa siihen, millainen järjestelmä täyttää asiakkaan vaatimukset. Vaatimuksista laaditaan vaatimusmäärittelydokumentti, johon tulee kirjata kaikki ominaisuudet, jotka asiakas tarvitsee. Haastavan vaiheesta tekee se, että asiakkaan todelliset tarpeet on tunnistettava ja ymmärrettävä jo tässä vaiheessa perusteellisesti. Tällaisia vaatimuksia kutsutaan usein asiakasvaatimuksiksi (customer - / business requirement) Mikäli esitutkimusvaiheessa asiakasvaatimukset ovat puutteellisia, virheellisiä tai väärin ymmärrettyjä, vaikuttaa se oleellisesti lopputulokseen. (Haikala & Märijärvi 2004, 37.)

Esitutkimusvaiheen perusteella siirrytään määrittelyvaiheeseen, jossa asiakasvaatimukset analysoidaan, josta johdetaan järjestelmävaatimukset (requirements analysis). Vaatimusmäärittelyiden pohjalta tehdään vielä toteutusmäärittelydokumentti, jossa kuvataan ohjelmiston toiminnot ja rajoitukset. Määrittelyvaihe edellyttää myös erityistä tarkkuutta, koska dokumenttia vastaan tarkastellaan lopuksi sisältääkö lopputuotos kaikki vaaditut ominaisuudet. Määrittelyvaiheen tarkoitus on kuvata ”mitä” järjestelmä tekee ja suunnitteluvaihe vastaa kysymykseen ”miten” se kyseisen ominaisuuden tekee. (Haikala & Märijärvi 2004, 38–40.)

Suunnitteluvaiheessa (design) ohjelmistovaatimusten toteutus kuvataan määrittelemällä järjestelmän tekniset vaatimukset, mikä pitää sisällään järjestelmän arkkitehtuurisen ja moduulien sisäisen rakenteen suunnittelun ja niiden tuloksena syntyvän dokumentaation, jonka jälkeen toteutusvaiheessa (programming) kehittäjät alkavat kirjoittaa ohjelmakoodia suunnitteludokumentin mukaisesti. Kun kaikki vaatimusmäärittelyssä kuvatut ominaisuudet on toteutettu, on se valmis testattavaksi. (Haikala & Märijärvi 2004, 40.)

Testausta (testing) tehdään toteutusvaiheen jälkeen ja sen tarkoitus on etsiä virheitä (bug) ohjelmistosta. Testausta voi tehdä monella tasolla. Testauksessa lopputuloksen oikeellisuutta verrataan määrittelyyn. Testauksessa löytyneet ”bugit” vaihtelevat järjestelmän kokonaan estävästä virheestä käyttäjää ärsyttävään yksityiskohtaan. Tulkinnoissa voi tulla erimielisyyttä: asiakkaan mielestä järjestelmässä esiintyvä epäkohta on selvä virhe, mutta toimittajan mielestä järjestelmään kuuluva ominaisuus. Testaaminen on tärkeä osa-alue ja siihen tulisi varata riittävästi aikaa. Heikko testaaminen voi aiheuttaa lopputuotoksessa ongelmia ja ikäviä yllätyksiä. (Haikala & Märijärvi 2004, 40, 287.)

Käyttöönottovaiheessa (release) ohjelmisto otetaan todelliseen käyttöön ja luovutetaan asiakkaalle ja ohjelmisto siirtyy ylläpitovaiheeseen. Ylläpitovaiheessa (maintenance) konsultoidaan asiakasta, pyritään ratkaisemaan asiakkaan ongelmia, korjataan ohjelmistossa esiintyviä virheitä ja käsitellään ohjelmiston parannusehdotuksia. Parannukset, korjaukset ja lisäykset toteutuvat seuraavassa projektissa ja saadaan ohjelmistoon seuraavan tuoteversion mukana. (Haikala & Merijärvi 2004, 40–41.)

### 3.1 Vesiputousmenetelmän hyödyt

Vesiputousmenetelmää on helppo käyttää ja hallita projektimenetelmänä, koska malli noudattaa aina samaa vaiheittaista kaavaa. Jokaisessa vaiheessa on alku- ja loppupiste ja jokainen vaihe on tarkoitettu toteuttamaan tiettyjä tuloksia ja tarkistuksia. Vesiputousmallissa noudatetaan kurinalaisuutta ja projektin edistymistä on helppo jakaa sidosryhmien ja asiakkaiden kanssa. Keskittymällä tarkasti vaatimuksiin ja toteutuksen suunnitteluun, tiimi vähentää epäonnistumisen riskiä aikataulun ja laadun suhteen. (Smartsheet: Agile methodology.)

Vesiputous edellyttää dokumentointia jokaista vaihetta varten, mikä parantaa koodien ja testien ymmärrettävyyttä. Dokumentit toimivat myös hyödyllisenä materiaalina tulevissa hankkeissa tai jos sidosryhmien on tarkasteltava yksityiskohtaisemmin tiettyä vaihetta. (Smartsheet: Agile methodology). Myös Mahalakshmi ja Sunarajajan toteavat, että vesiputousmallin hyötynä voidaan pitää tarkkaa hyvin määriteltyä dokumenttia. Mikäli se on laadukkaasti laadittu ja kaikki asiakkaan tarpeet otettu huomioon, seuraa siitä myös laadukas lopputuotos käyttöönottovaiheessa. (Mahalakshmi & Sunarajajan 2013, 1).

Myllymäki ym. (2015, 85) toteavat, että vesiputousmallin vahvuus on huolellisessa suunnittelussa, koska virheet ovat helpompi ja halvempi korjata suunnitteluvaiheessa kuin vasta ylläpitovaiheessa. Vesiputousmalli sopii sellaisille ohjelmistoille, joiden vaatimukset voidaan jo alkuvaiheessa määrittää tarkasti etukäteen ja voidaan luottaa siihen, ettei muutoksia suunnitteluun ja toiminnallisuuteen tule ja virheiden mahdollisuus on pieni prosessin myöhemmissä vaiheissa (Smartsheet: Agile methodology). Vesiputousmallin hyötynä on yksikertaisuus ja selkeys (Juvonen 2018, 15).

Mikäli kyseessä on pieni rajoittunut ohjelmisto ja jokaisella projektiin osallistuvalla on tarkka ymmärrys sen yksityiskohdista, projekti vesiputousmallilla voi onnistua hyvin. Jos ongelmia ilmaantuu, voi yksikin projektinjäsen tai yksikin lisäresurssi pelastaa

projektin vaikeuksien ilmaannuttua. (Haikala & Märijärvi 2004, 30). Myös Mahalakshmi ja Sunarajan (2013, 1) mukaan vesiputousmallin toteuttamiseen tarvittavien resurssien määrä on vähäinen. Suurta projektia, joka on ajautunut ongelmiin ei kuitenkaan pelasteta yksittäisellä urotyöllä tai henkilökunnan lisäämisellä. Ihmisten lisääminen aikataulusta myöhässä olevaan projektiin viivästyttää sitä entisestään (Haikala & Märijärvi 2004, 30).

### 3.2 Vesiputousmallin haitat

Tarkkojen vaatimusten kerääminen voi olla haastavaa. Yksi vesiputoushankkeen ensimmäisistä vaiheista on keskustella asiakkaiden ja sidosryhmien kanssa ja tunnistaa heidän vaatimukset. Voi kuitenkin olla vaikeaa määrittää tarkalleen, mitä kaikkea asiakkaat haluavat hankkeen alkuvaiheessa. Usein asiakkaat eivät tiedä, mitä he haluavat alkuvaiheessa, vaan oppivat ja tunnistavat vaatimukset projektin edetessä. (Smartsheet: Agile Methodology.)

Vesiputousmallin suurin haittapuoli on se, miten se käsittelee muutosta. Koska vesiputousmalli on lineaarinen, peräkkäinen malli, ei vaiheiden välillä voi hyppiä, vaikka odottamattomia muutoksia matkan varrella tapahtuukin. Esimerkiksi kun saavutetaan testausvaihe ja siinä kohtaa huomataan, että jokin ominaisuus puuttui vaatimusmäärittelystä, on erittäin vaikeaa ja kallista mennä takaisin alkupäähän ja korjata se (Smartsheet: Agile Methodology.)

Myös Juvosen 2018 mukaan vesiputousmallin huonona puolena on, ettei menetelmä toimi, koska usein tosielämässä tulee tarve palata aikaisempiin vaiheisiin ja muuttaa suunnitelmia ja toteutusta (Juvonen 2018, 16.) Vesiputousmallin kritiikkinä esitetään, että uskotaan olevan mahdotonta suunnitella täydellisesti etukäteen ohjelmistoa niin, ettei aikaisempiin vaiheisiin tarvitsisi palata. Usein asiakas muuttaa vaatimuksiinsa kesken projektin ja tämä tarkoittaisi sitä, että kaikki suunnitteluun käytetty aika menisi hukkaan ja jouduttaisiin aloittamaan alusta. Perinteisessä lähestymistavassa



kehittäjien tulee ymmärtää vaatimukset tarkasti ennen siirtymistä seuraavaan vaiheeseen. Nykyisessä muuttuvassa maailmassa vaatimukset saattavat kuitenkin olla epäselviä tai jopa tuntemattomia. Markkinoita ei voi ennalta määrittää ja saattaa olla, ettei kukaan ymmärrä vielä selkeästi kehitteillä olevaa tuotetta. (Rising & Norman 2000, 26.)

Mahalakshmi & Sunarajajan (2013) mukaan jokaisessa vaiheissa tulee usein ongelmia ja mikäli niitä ei ratkaista kokonaan ja siirrytään siitä huolimatta seuraavaan vaiheeseen, edessä on ongelmia. Isojen kokonaisuuksien aikatauluarviointi on vaikeaa ja ne usein pettävät. Standish Chaos 2012 raportin mukaan myös Jeff Sutherland on todennut, että työskennellessään vesiputousmallin mukaisissa projekteissa, projektit olivat usein myöhässä eivätkä johtajat olleet tyytyväisiä. Ihmisiä asetettiin suuren paineen alle, mikä myös turhautti ihmisiä. Mitä enemmän painetta asetettiin, sen haastavammaksi työ kävi. (Chaos Manifesto 2012, 32.)

Schatz ja Abdalshafi (2005) mukaan Primavera Systems nimisen yhtiön, joka tarjoaa yritysprojektien hallintaratkaisuja, työskentelyyn kuului pitkään yleisesti siihen aikaan käytössä oleva vesiputousmalli ja heidän tiimensä työskenteli pitkiä työpäiviä, usein myös viikonloppuisin tuottaakseen laadukkaita julkaisuja sovitussa aikataulussa. Julkaisun jälkeen he juhlistivat omasta mielestään onnistunutta julkaisua. Pian julkaisun jälkeen kävi kuitenkin ilmi, että julkaisun laadusta oli erilainen käsitys kehityksen ulkopuolella. Muiden mielestä uudesta versiosta puuttui ominaisuuksia ja sieltä löytyikin laatu sekä käytettävyyso ongelmia. Tämä luonnollisesti toi turhautumisen ja pettymyksen tunteita ja yleistä alakuloa tiimin ympärille. (Schatz & Abdalshafi 2005, 36–37.)

Rico, Hasan, Sayani ja Sutherland toteavat, että jo 1950-luvulla perinteinen menetelmä on ollut periaatteiden, standardien, prosessien, menettelyjen dokumentaation, työkalujen ja mittareiden yhdistelmä ja tällä on pyritty takaamaan ohjelmiston laatu. Toisin sanoen laadun uskotaan tulleen suurten prosessien funktiosta. Perinteiset menetelmät ovat kuitenkin saavuttaneet pisteen, jonka mukaan tuotto on vähentynyt, mikä on vaikuttanut haitallisesti niiden arvolupaukseen. Kustannukset ovat nousseet

merkittävästi ja perinteisten mallien tuomat edut ovat liian alhaiset. (Rico, Sayani & Sone 2009, luku 23.)

Huolimatta vesiputousmallin useista haitoista sillä katsotaan kuitenkin olevan niin paljon hyötyjä, että se pitää pintansa edelleen suosittuna projektinhallintamenetelmänä ohjelmistonkehitysprojekteissa (Mahalakshmi & Sunarajan 2013, 3)

## **4 Ketterä kehitys eli Agile**

### **4.1 Ketterän kehityksen periaatteet**

90-luvun puolivälissä on alettu hahmottelemaan vaihtoehtoista lähestymistapaa perinteiselle ohjelmistokehitykselle. Hughesin 2016 mukaan The Standish Group oli julkaissut tuolloin kyselynsä tulokset, johon oli vastannut 8380 henkilöä, jotka työkentelivät 365 erilaisessa projektissa. Kyselytutkimuksen myötä paljastui, että valtaosa perinteisen lähestymistavan projekteista olivat epäonnistuneet aikataulun, kustannusten ja luvattujen järjestelmän ominaisuuksien suhteen. (Hughes 2016, 13)

Vastavetona vesiputousmallin kankeudelle voidaan pitää ketteriä menetelmiä. Juvosen 2018 mukaan vuonna 2001 joukko ohjelmistokehittäjiä oli kokoontunut keskustelemaan käyttämistään menetelmistä ja päättivät nimittää parhaita menetelmiä ketteriksi ja perustivat järjestön Agile Alliance, joka edistää ketterien menetelmien käyttöä (Juvonen 2018, 17.) Agile Manifestossa (2001) on määritelty ketterien menetelmien pohjana olevat arvot ja keskeiset periaatteet. Agile Manifesto-julistuksen ketterien menetelmien arvot ja periaatteet löytyvät liitteestä 1.

Ketterän kehityksen projektit ovat kasvaneet viimeisten kymmenen vuoden aikana huomattavasti. Joko Scrum, Lean tai XP tai muu ketterän kehityksen metodi on ollut käytössä ohjelmistonkehitysprojekteissa (Mersino 2013, 242.)

Ketterän kehityksen ajatusmalliin kuuluu, että ei ole olemassa vain yhtä oikeaa tapaa saavuttaa haluttu lopputulos. Harvoin ketterissä menetelmissä tarkkaan määritetään sääntöjä ja ohjeistuksia siitä, kuinka missäkin tilanteessa tulisi toimia. On olemassa kaikille ketterille malleille yhteisiä yleisperiaatteita:

- asiakastyytyväisyys
- muutosten hyväksyminen
- julkaise aikaisin ja usein
- toimiva sovellus
- tasainen työtahti
- tiivis yhteistyö toteuttajien ja liiketoiminnan edustajien välillä
- suora keskustelu
- luottamus tekijöihin
- tekninen loistokkuus
- yksinkertaisuus
- itseohjautuvuus
- itsetarkastelu (Ketteryys haltuun: Yleisimmät ketterät käytännöt.)

Alistair Cockburn, joka on osallistunut Agile Manifeston määrittämiseen, on tarjonnut merkityksen ketterälle kehitykselle.

*A system of methods designed to maximize the alignment of the work done by the developers with the direction needed by the business at the time, especially in a context where the business directions changes frequently...[where] important facts change, or where we are obliged to adapt to important uncontrolled factors.*

(Auer, Auer, Heinäsmäki, Hölttä, Kalliala, Laanti, Laine, Lekman, Miinalainen, Naski, Piiparinen, Puhakka, Pyhäjärvi, Pääkkönen, Räisänen, Sora, Taipale, Talvio, Tanninen, Toikkanen, Toivola, Toro, Valsta, Väyrynen & Weissenberg 2013, 26).

Eli ketterän kehityksen tarkoituksena on maksimoida kehittäjien tekemä työ liiketoiminnan haluamaan suuntaan erityisesti tilanteissa, jossa suuntaviivat ja tarpeet

muuttuvat usein ja olemme pakotettuja hallitsemaan tärkeitä odottamattomia ja hallitsemattomia tekijöitä.

Ketterä kehittäminen on kehittämismalli, joka soveltuu erityisen hyvin kehittämiseen, jossa on tarve saada jatkuvaa palautetta käyttäjiltä kehitettävästä kokonaisuudesta. Ketterän kehittämisen periaate on iteratiivisuus, mikä tarkoittaa, että kehittäminen pilkotaan osiin, jota kehitetään lyhyissä kehitysjaksoissa tavoitteena saada jatkuvaa palautetta käyttäjiltä ja asiakkailta. (Suomidigi: Ketterän kehittämisen johtaminen.)

### **Ketterän kehittämisen hyödyt**

Tänä päivänä liiketoiminnalle arvokkaita asioita ovat mm.

- läpimenoaika
- maksavien asiakkaiden tyytyväisyys
- laatu
- läpinäkyvyys, josta seuraa ohjattavuus ja ennustettavuus (Auer ym. 2013, 26).

Perinteisesti kehittämistä ohjataan seuraamalla toteutuneita kuluja ja valmiusasetta. Ketterässä kehityksen ohjauksessa painopiste siirtyy menneisyyden seuraamisesta tulevaisuuden ennustamiseen. Toimivaa ohjelmistoa syntyy jo varhain pienemmissä palasissa ja näin ollen todellinen tilanne on tiedossa eikä tarvitse arvailla, missä vaiheessa ominaisuuksien kehittäminen on. Ketterä kehittäminen mahdollistaa siirtymisen kulujen seurannasta arvon seuraamiseen. (Auer ym. 2013, 28.)

Ketteryyden perusidea on tuottaa asiakkaan tarpeisiin laadukas ohjelmisto nopeammin ja vähemmin resurssein. Tätä perusajatusta eri ketterät menetelmät ja käytännöt pyrkivät edistämään. Tästä johtuen organisaation asiakasfokus ja asiakkaan todellinen ymmärtäminen organisaatiossa on hyvin tärkeää. Organisaation tuleekin pyrkiä pois käsky-vastuu-raportointi-kulttuurista kohti kulttuuria, jossa asiakkaan kuunteleminen, asiakkaan tarpeiden ymmärtäminen ja niihin vastaaminen ovat avainasemassa. (Auer ym. 2013, 43.)

## Johtajuus ketterässä kehittämisessä

Ketterän kehittämisen johtamisella voidaan panostaa koko organisaation kannalta oikeisiin asioihin, mikä parantaa kehittämisen vaikuttavuutta. Ketterän kehittämisen tuottavuus saadaan enemmän näkyväksi ja tuottavuutta voidaan jatkuvasti parantaa johtamismallin avulla. Kehittämisen kokonaislaatu ja ratkaisujen yhteentoimivuus myös paranevat. (Suomidigi: Ketterän kehittämisen johtaminen.)

Mersinon 2013 mukaan jokainen, joka on ollut mukana ketterissä projekteissa, on varmasti sitä mieltä, että ketterän tiimin johtaminen ei ole samanlaista kuin perinteisten tiimien johtaminen. Ketterien tiimien odotetaan olevan itseohjautuvia, mikä ei tarkoita sitä, etteikö tiimiä johdettaisi, vaan että se vaatii johtamiselta vähemmän ohjaamista ja enemmän kannustamista ja rohkaisemista mikä taasen edellyttää johtajalta enemmän tunnetaitoja ja -älyä. Itseohjautuvissa tiimeissä johtajan tehtävä on tukea tiimiä itseohjautuvuuteen siten, että jäsenet tietävät itse mitä tehdä seuraavaksi. (Mersino 2013, 242, 250.)

Usein projektipäälliköillä on valtavasti erilaisia tehtäviä. He jakavat töitä ja vastuita projektinjäsenille, he ohjaavat toimintaa ja keräävät edistymisstatistiikkaa ja erilaisia raportteja sekä päivittävät projektin tilannetta ja edistymistä yritysjohdolle. Itseohjautuvat tiimit toimivat toisin, koska suurimman osan edellä mainituista toiminnoista tiimi hoitaa itse. Joka tapauksessa tiimin jäsenet eivät tarvitse kontrolloijaa ja henkilöä kertomaan, mitä heidän pitää tehdä. (Mersino 2013, 243.)

Joskus muiden johtajien organisaatiossa voi olla vaikea sopeuttaa oma tyylinsä johtaa ketteriin menetelmiin. Tavat, joilla perinteisesti on totuttu johtamaan, eivät usein toimi ja johda menestykseen ketterissä tiimeissä. Ongelmia voi tulla esimerkiksi päivittäisten kokoontumisien yhteydessä, jos johtaja haluaa pitää ne aikaisemman käytännön mukaisesti kaikkien kanssa erikseen ja kuulla mitä kukin on saanut aikaan ja kontrolloida siten tehtävien edistymistä. Johtaja toimii tällöin tiedonjakokeskuksena ja hänelle jää kaikki tieto, mitä ympärillä tapahtuu, kun sitä vastoin Scrum-tapojen mukaisesti päivittäiset kokoontumiset pitäisi pitää yhdessä ja keskustella asioista tiimin kesken, jolloin kaikilla on tieto missä mennään asioiden kanssa. Johtajan tulee

huolehtia siitä, että päivittäiset kokoontumiset fokusoituvat siihen, että tiiminjäsenet keskustelevat ja heidän itsensä tulisi pitää vaimeampaa ääntä. (Mersino 2013, 246.)

Ohjelmistot ovat jo pitkään olleet niin monimutkaisia, ettei yhdet aivot eivät enää riitä jäsentämään sen kaikkia yksityiskohtia. Ratkaisujen tekemiseen tarvitaan tiimejä tai jopa useampia tiimejä. Koska ohjelmisto halutaan toteuttaa ja julkaista toiminnallisuus kerrallaan muodostetaan myös sen ympärille tiimejä, jotka voivat toteuttaa toiminnollisuuksia yksi kerrallaan. Sen sijaan, että yritysjohto ratkoisi toiminnollisuuksien välisiä riippuvuuksia, ketterässä kehityksessä suositaan, että suunnittelijat keskustelisivat toistensa kanssa ja sopisivat riippuvuuksista. Johto tuskin on koskaan sellillä toteutuksen yksityiskohdista samalla tasolla kuin itse toteuttajat. (Auer ym. 2013, 10–11). Organisaation johdossa saatetaan yllättyä, kun kehitystiimi tai sen jäsen voi käyttää oma päätösvaltaansa saavuttaakseen tavoitteen. Yllätyksen tunne häviää pikkuhiljaa, kun huomataan, että tuottavuus ja itseohjautuvuus kasvavat. (Schwaber & Beedle 2002, 38.)

Kun puhutaan itseohjautuvuudesta saattaa usein esiintyä väärää kuvitelmaa, että itseohjautuvuus tarkoittaa, että tiimin jäsenet voivat tehdä, mitä heitä huvittaa ja ettei johtajuudelle olisi enää käyttöä. Itseohjautuvat tiimit eivät vapaudu johtamiselta. Edelleen johto päättää suuret linjaukset esimerkiksi mitä tuotetta kehitetään ja valitsee henkilöt, jotka työskentelevät projektissa tuotteen ympärillä ja tekevät päätöksiä asioista, joihin myös tiimeillä on vaikutus. Scrum-tiimin tehtävänä on itseohjautua haasteiden ympärille niiden rajojen sisällä, jotka johto on heille asettanut. Mikäli johtajat ohjaavat liikaa sitä, miten tiimi ratkaisisi esiin tulleet haasteet, itseohjautuvuutta ei tapahdu. Johdon tehtävä on poistaa esteet itseohjautuvuuden tieltä. (Cohen 2010, 220–221.)

## 4.2 Siirtyminen ketteryyteen

Organisaation on pystyttävä valitsemaan, mitä ketterien menetelmien tavoitteita, periaatteita ja käytäntöjä se ryhtyy noudattamaan, koska suoraviivaista vastausta tähän ei ole. Organisaatio toimivat eri ympäristöissä ja ovat eri kehitysasteilla ja menetelmien käyttöönotto riippuu aina organisaatiosta tai sen kontekstista. Jos organisaatio yrittää toteuttaa liian edistyneitä menetelmiä tai liian radikaaleja muutoksia, epäonnistumisen riski on suuri. Menetelmien käyttöönotto vaatii kaikkien osapuolten aitoa ymmärrystä menetelmän tarkoituksesta sekä ehkäistä muutosvastarintaa, joka syntyy halusta pysytellä mukavuusalueella. Yksi tärkeimmistä johtamishaasteista on vallan ja vastuun, sekä luottamuksen antaminen tiimitason ihmisille. Muutos ei saa olla liian jyrkkä eikä myöskään liian epämääräinen, tai sillä ei päästä toivottuun lopputulokseen. (Auer ym. 2013, 43–44.)

Rugbyn ja muiden (2018) mukaan olosuhteet ovat kypsiä ketterille tiimeille missä tahansa tilanteessa, jossa ongelmat ovat monimutkaisia, ratkaisut ovat aluksi epäselviä, projektivaatimukset tulevat todennäköisesti muuttumaan ja missä tiivis yhteistyö loppukäyttäjien kanssa on mahdollista. Ketteriä menetelmiä on otettu käyttöön ensimmäiseksi IT-osastoilla ja niitä käytetään laajasti ohjelmistokehityksessä. Ajan mittaan ne ovat levinneet myös muuhun tuotekehitykseen, markkinointiin ja jopa HR-toimintoihin (Rugby, Sutherland & Nobel 2018.)

Siirtyminen perinteisistä malleista ketterimpiin menetelmiin organisaatiossa kannattaa aloittaa pilottiprojektilla sekä oppia siitä mahdollisimman paljon ja levittää ketterää menetelmää läpi koko organisaation. Uudet ketterät tiimit hyötyvät ensiksi perustetun ketterän tiimin opeista. Syitä siihen, että organisaatiossa otetaan ketterä malli käyttöön ensin yhden tiimin kokeiluna, on monia. Se on halvempi tapa kuin, että koko organisaatiossa muutettaisiin kaikki tiimit ketteriksi kerralla kaikkien opetellessa uusia toimintatapoja yhtä aikaa monien Scrum Mastereiden ja kouluttajien ohjauksessa. Myös ensivaiheessa mahdollisesti tehdyt virheet vaikuttavat vain yhteen tiimiin eikä koko organisaatioon. (Cohn 2010, 44.)

Olisi myös stressaavaa koko organisaatiolle, jos työntekijöille ilmoitettaisiin, että kaikki siirtyvät yhdessä tekemään asioita uusilla työtavoilla. Valitsemalla Scrum-tiimin jäsenet huolellisesti sekä pilottiprojektin, jolla Scrumia ensi kertaa toteutetaan, takaa melko varman projektin onnistumisen. Lisäarvoa tuo, kun tiimi menestyy ja oppii kokemuksistaan ja tätä kautta voidaan osoittaa epäluuloisten ja skeptikoiden luottamus, mikä on tärkeää Scrumin laajentamisessa ja jalkauttamisessa koko organisaatioon. Koko organisaation siirtyminen toteuttamaan Scrumia kerralla vaatisi jonkin asteista uudelleen järjestäytymistä organisaatiossa, mikä saattaisi herättää vastustusta joissakin työntekijöissä. (Cohn 2010, 45.)

Myös sillä, että koko organisaatiossa ryhtyisi kaikki kerralla toteuttamaan Scrumia, on hyötyjä. Jonkin verran vaikeuksia saattaa aiheuttaa se, että yksi tiimi toimii ketterän kehityksen periaatteiden mukaisesti ja tekevät yhteistyötä tiimin kanssa, joka toimii perinteisten oppien mukaan. Haasteita voi tulla tiimien välisten erilaisten työskentelytapojen takia suunnittelun, kommunikaation ja aikataulujen suhteen. Tämänkaltaiset ongelmat poistuvat, kun kaikki omaksuvat Scrumia yhtä aikaa. (Cohn 2010, 46.)

Organisaatio, jonka pyrkimyksenä on ketteröityä, saattaa törmätä ongelmiin, jotka johtuvat ketteryyden puutteellisesta ymmärryksestä tai soveltamiskyvystä asiakkaan puolella. Oma organisaatiota tällaisessa tilanteessa kannattaa ketteröittää, jotta voidaan avata ketteryyden suomia etuja asiakkaille. (Auer ym. 2013, 43.)

Rigby ja muut (2018) ovat tutkineet ketterien menetelmien skaalautumista sadoissa yrityksissä. Suuremmat yritykset kuten Spotify ja Netflix ovat syntyneet ketterinä ja saavuttaneet huikkeen kasvun. Mutta heidän tutkimus osoitti myös, että menestystarinoiden ohella on myös koettu pettymyksiä organisaatioissa, joissa ketteryyttä on laajennettu. On ajateltava ketteryyden laajentamista realistisesti, kaikkia toimintoja ei voida järjestää ketteriksi.

Hammarstenin (2016) artikkelissa kerrotaan, että Aalto yliopiston tutkija Hiekkänen myös huomauttaa, että toimintatapa vaatii osaamista eikä organisaation siirtyminen ketteryyteen välttämättä käy kovin ketterästi, koska yritysten tilanteet, valmiudet ja



toimintakulttuurit ovat erilaisia. Ketteryyttä voi kokeilla yksittäisellä projektilla, mutta malli vaatii pitkää sisäänajoaikaa ja syvällistä menetelmän tuntemista. *Jos ihmisille annetaan valta ilman osaamista, syntyy sekasotku. Kaikkien osapuolten tulee ymmärtää mihin ollaan ryhtymässä* (Ketteryys ei sovi kaikkeen 2016.)

## 5 Scrum projektinhallintamenetelmänä

Scrumin mukaan isot tarkkaan suunnitellut projektisuunnitelmat eivät toimi, koska ohjelmistokehitys on tänä päivänä arvaamatonta ja siksi tarvitaan mukautettua ohjelmistokehitystapaa. Scrum on syntynyt kahdesta syystä: aikanaan olemassa olleet menetelmät eivät olleet toimivia ja projektien onnistumiseksi tarvittiin uusi menetelmä. Scrumin ensisijaisena tavoitteena on menestyvä projekti. (Rico ym. 2009, luku 5.1.)

Scrum ei ole prosessi, tekniikka eikä yksityiskohtia ohjaava menetelmä vaan viitekehys, joka on tarkoitettu monimutkaisten tuotteiden kehittämiseen, toimittamiseen ja ylläpitoon. Scrumin määritelmä koostuu rooleista, tapahtumista, tuotoksista ja säännöistä, jotka sitovat Scrumin elementit yhteen ja jokainen elementti palvelee tiettyä tarkoitusta. Viitekehyksen avulla voidaan ratkaista monimutkaisia ongelmia kehitettäessä mahdollisimman suurta lisäarvoa tuotteita luovasti ja tuottavasti. Scrum on

- kevyt
- yksinkertainen ymmärtää
- vaikea taitaa. (Schwaber & Sutherland 2017, 3.)

Ken Schwaber on ollut kehittämässä Scrum-prosessia ohjelmistokehitystä varten. Hän oli oivaltanut ollessaan asiakkaansa luona, miksi jokaisella, joka on ollut mukana ohjelmiston kehityksessä, on ongelmia sen toteuttamisessa. He kaikki tuhlasivat aikaa yrittämällä hallita isoa kuvaa, isoja kokonaisuuksia ja yhteistä linjausta, kun ainoa kunnollista hallinnan tunnetta toi usein tapahtuva ensi käden tarkastus ja minkä seurauksena tuli usein ilmi välittömiä muutostarpeita. Tähän näkemykseen perustuen Schwaber on alkanut kehittää lyhyempiin kehitysjaksoihin pohjautuvaa Scrum-

prosessia monimutkaiseen tuotekehitykseen, erityisesti ohjelmistokehitykseen. (Schwaber & Beedle 2002, 19,25.)

Scrum on maailmalla parhaiten tunnettu ketterä menetelmä (Juvonen 2018, 19) ja tärkein ketterän kehityksen menetelmistä Agile Manifesto-koulussa. Viimeisten mitausten mukaan yli puolet ketteristä projekteista käyttää Scrumia projektinhallintamenetelmänä. Scrum on osoittautunut menetelmäksi, jonka omaksuu nopeasti ja jonka avulla kommunikointi, projektin seuraaminen helpottuvat ja Scrumin räätälöinti omiin tarkoituksiin on sujuvaa. Tämä tekee Scrumista helposti opeteltavan menetelmän järjestelmäasiantuntijoille, joille iteratiivinen kehittäminen on uutta. (Hughes 2016, 20.)

Pienimmillään Scrum on empiirinen tutkimusmalli, mikä tarkoittaa, että tieto saadaan kokemuksesta ja päätökset tehdään kokemuksen perusteella sen sijaan, että simuloidaan päätöksiä perustuen tarkkoihin tutkimuksiin ja matemaattisiin kaavoihin. Scrumissa voi pilkkoa projektin käytännöllisiksi paloiksi ja seurata tuloksia askel askeleelta. Oli kyse sitten uuden älypuhelimien julkaisemisesta tai tyttären 5-vuotissyntymäpäivien organisoimisesta; Jos haluat saada missä tahansa projektissa valmista, Scrum tarjoaa rakenteen tehokkaampien ja nopeampien tulosten aikaansaamiseksi. (Layton 2015.)

Scrumin tapa toimia on täysin erilainen kuin muissa ohjelmistonkehitysprojektimenetelmissä. Scrum näyttää ja tuntuu erilaiselta, koska se perustuu empirismiin. Vähemmän aikaa käytetään yrittäessä tarkkaan suunnitella ja määrittellä tehtäviä. Vähemmän aikaa käytetään dokumenttien laatimiseen ja lukemiseen. Enemmän aikaa käytetään tiimin kesken ymmärtämiseen mitä tapahtuu ja kokemusperäiseen palautteeseen. Useimmat ihmiset ymmärtävät Scrumin yksinkertaisuuden ja tuottavuuden vasta sen jälkeen, kun ovat alkaneet käyttää menetelmää ja samalla he ymmärtävät, kuinka sopimattomia perinteiset mallit ovat ohjelmistonkehitysprosessiin. (Schwaber & Beedle 2002, 26.)

Yksi Scrumiin siirtymisen tavoitteista on saada koko tiimi työskentelemään sen eteen, että saadaan julkaistua loistava tuote ja kaikista huonoista tavoista, mitkä ovat vastaan tätä tavoitetta, pyritään pääsemään eroon. Kirjoitetut dokumentit estävät tiimin yhteisöllisyyden tunnetta, koska kaksisuuntainen kommunikointi on niiden osalta masentavaa. Yksi tiimin jäsen sanoo ”tässä on kirjoitettu mitä pitää tehdä” ja muiden tiimin jäsenten odotetaan tekevän tarkalleen mitä on kirjoitettu. Tällainen isäntä- ja renkisuhde ei todennäköisesti luo vahvoja sitoutumisen tunteita tiimin jäsenissä. Sen sijaan, että he tuntisivat vastuuta tuotteen onnistumisesta, he tuntevat olevan enemmän vastuussa siitä, mitä on kuvattu dokumentissa. (Cohn 2009, 237.)

Voi olla vaikea päästä eroon tarkasta dokumentoinnista, jos on tottunut tekemään projektissa tarkkoja vaatimuksia esimerkiksi ISO9001 standardin mukaisesti, mikä perustuu tarkkaan dokumentointiin ja jäljittämiseen. Kaikkea dokumentaatiota ei tarvitse heittää pois vaan Scrumin mukaan tarkoituksena on karsia kaikki turha pois ja säilyttää dokumentit niin tiivistettynä kuin mahdollista. (Cohn 2009, 238.)

Vaikka Agile Manifesto (Beck ym. 2001) julistetaan, että ”arvostamme toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota” se ei tarkoita sitä, että ketterä kehitys vastustaa kaikkea dokumentaatiota ja että jo kirjoitetut dokumentit tulisi hylätä täysin. Tavoite ketterässä kehityksessä on löytää oikeanlainen suhde dokumentoinnin ja kommunikaation välille. Olemme olleet vain liikaa kallellaan tarkan dokumentoinnin puolelle. On hyvä myös ymmärtää, että vaatimusmäärittely on vain yksi dokumentoinnin muoto. Esimerkiksi myös testisuunnitelmat, suoritettavat testitapaukset ja jopa ominaisuutta varten ohjelmistoon tehty koodi toimivat dokumentaationa. (Cohn 2009, 238.)

## 5.1 Scrumin roolit

### **Tuoteomistaja (Product Owner)**

Scrum istuttaa liiketoiminnan edustajan ohjelmiston kehitystiimiin. Tämän liiketoiminnan edustajan tulee ymmärtää hyvin liiketoiminnan tarpeita, koska Scrum-tiimi

vastaanottaa suurimman osan ohjeista ja suuntaviivoista tältä yksilöltä. Tiimi on riippuvainen hänen päätöksistä sen suhteen, mitä ominaisuuksia ohjelmistoon tuodaan ja missä järjestyksessä eli hän on vastuussa kehitysjonon hallinnasta ja valvonnasta. Liiketoiminnan edustajan tarkastaessa tiimin tuotoksia jokaisen iteraation lopussa, hän myös hyväksyy tai hylkää moduulit perustuen siihen, kuinka hyvin ne vastaavat liiketoiminnan tarpeita. Koska tämän liiketoiminnan edustajan päätökset heijastuvat vahvasti tuotteen lopputulokseen, tätä yksilöä kutsutaan tuoteomistajaksi. (Hughes 2016, 20.)

Tuoteomistaja on yksi henkilö, ei ryhmä eikä komitea. Tuoteomistaja vastaa tuotteen arvon maksimoimisesta, mikä kehitystiimin tuotoksena saadaan aikaan. Se, miten tämä toteutetaan, voi vaihdella merkittävästi organisaatioiden, Scrum-tiimien ja yksilöiden välillä. (Schwaber & Sutherland 2017, 7.)

Juvosen 2018 mukaan tuoteomistaja edustaa asiakasta. Hänen tehtävänä on laatia tuotteen työlista, ylläpitää työlistaa ja määrittää toteutettaville ominaisuuksille prioriteetti. Perusedellytys tuoteomistajan toimimiselle on tehdä tuotetta koskevia päätöksiä. Tuoteomistajalla ei tarvitse välttämättä olla ohjelmistoalan osaamista ja ymmärrystä teknisistä asioista. (Juvonen 2018, 19). Usein tuoteomistajana toimii henkilö, jolla on usean vuoden työkokemus ja hän on toiminut kehitettävän järjestelmän käyttäjänä pitkään (Cohn 2009, 130).

Tuoteomistaja vastaa myös siitä, että tuotteen työlista on läpinäkyvä kaikille. Jokaisen projektissa työskentelevän on oltava tietoinen, mitkä ominaisuudet ovat korkeimmalla prioriteetilla työn alla. Mikään ryhmä tai yksittäinen henkilö ei voi muuttaa tuotteen työlistan prioriteettijärjestystä vakuuttamatta ensin tuoteomistajaa. Jokaisen henkilön organisaatiossa on kunnioitettava tuoteomistajan päätöksiä. Kenelläkään ei ole lupa ohjata Scrum tiimiä tekemään muutoin kuin tuoteomistajan hyväksymän prioriteettijärjestyksen mukaisesti eikä Scrum-tiimi saa ottaa vastaan ohjeita muilta taholta. Kaikki tuoteomistajan päätökset ovat näkyviä, koska ne perustuvat tuotejonon priorisointiin. (Schwaber & Beedle 2002, 34.)

Scrum kehottaa painokkaasti, että tuoteomistajan tulee työskennellä lähellä kehitystiimiä sen sijaan, että tuoteomistaja osallistuu työskentelyyn etänä. On tärkeää, että tuoteomistaja istuu lähellä kehitystiimiä, jotta hän voi vastata kehittäjien kysymyksiin reaaliajassa kasvotusten liittyen määrityksiin. Tämä myös mahdollistaa tuoteomistajan validoida ja hyväksyä sovelluskomponentit heti kun kehittäjät ovat saaneet ne valmiiksi. (Hughes 2016, 20.) Myös Auerin ja muiden (2013, 36) mukaan koko Scrum-tiimin mukaan lukien tuoteomistajan on työskenneltävä samoissa tiloissa puhe-etäisyydeltä.

On erittäin tärkeää, että tuoteomistaja ymmärtää syvällisesti liiketoimintaa, markkinoita, asiakkaita ja loppukäyttäjiä ja heidän tarpeitaan ja hänellä on kykyä toimia eri sidosryhmien kanssa. Ammattitaitoinen tuoteomistaja jakaa tietoa eri sidosryhmille ja hän kuuntelee heitä, erityisesti hänen tulee kuunnella tiimin jäseniä. Tiimin jäsenet oppivat tuoteomistajalta enemmän kehitettävästä tuotteesta ja liiketoiminnasta, jolloin myös tiimi osaa tuoda esiin liiketoiminnan kannalta kriittisiä tekijöitä ja osaa suositella parhaita teknisiä ratkaisuja. Vaikka tuoteomistajalla onkin valtuudet priorisoida tuotteen työlistaa, hänen tulee kuitenkin kuunnella tiimin jäsenten suosituksia ja muuttaa tarvittaessa priorisointijärjestystä esimerkiksi tiimin esiintuomiin teknisiin tekijöihin perustuen. (Cohn 2009, 130–131.)

### **Scrum Master**

Scrum Master on Scrum-tiimin palveleva johtaja, joka vastaa Scrumin edistämisestä ja hänen tehtävä on auttaa Scrum-tiimiä toteuttamaan Scrumia. Scrum Master toteuttaa tämän auttamalla kaikkia ymmärtämään Scrumin teoriaa, käytäntöjä, sääntöjä ja arvoja. Scrum Master (Schwaber & Sutherland 2017, 7.) Cohn (2010, 118) kertoo, että Scrum Master on tiimin palveleva johtaja ilman auktoriteettia, Scrum Masterilla on auktoriteettia vain Scrum-prosessista.

Cohnin 2010 mukaan hyvällä Scrum Masterilla on oltava kuusi erityispiirrettä:

- Vastuullisuus, mikä ei tarkoita, että Scrum Master vastaisi koko tiimin menestyksestä, vaan tiimi vastaa siitä kokonaisuudessaan. Scrum Master vastaa tiimin suoritustehon maksimoinnista ja tiimin Scrumiin sopeuttamisesta ja toteuttamisesta.
- Nöyryys, mikä tarkoittaa, että Scrum Master ei ota kunniaa itselleen tiimin menestyksestä vaan siitä, miten auttaa tiimiä saavuttamaan menestystä. Scrum Masterin tulisi tehdä kaikki tarvittava, että tiimi saavuttaisi tavoitteensa. Hyvä Scrum Master tunnistaa jokaisen tiimin jäsenen arvon ja ohjaa myös muut ajattelemaan samoin.
- Yhteistyökyky. Scrum Masterin tulee varmistaa, että tiimin jäsenet voivat nostaa asioita esiin. Scrum Master rohkaisee tiimin jäseniä löytämään jokaista hyödyttävään ratkaisun sen sijaan, että tilanteessa syntyisi voittajia ja häviäjiä.
- Sitoutuminen. Scrum Masterin tulee sitoutua samalla tasolla projektiin ja Sprintin tavoitteisiin kuin tiimin jäsenten. Tapa, jolla Scrum Master voi osoittaa sitoutumistaan, on pysyä projektissa sen alusta loppuun.
- Vaikuttavuus. Scrum Masterin tulee tietää, kuka organisaatiossa tekee päätöksiä, kuinka päätöksiä tehdään, minkälaisia liittoutumia organisaatiossa on ja pyrkiä vaikuttamaan asioihin, mistä on hyötyä koko tiimille.
- Asiantuntemus. Scrum Masterin ei välttämättä tarvitse olla markkinointi guru tai ohjelmoinnin asiantuntija, mutta heillä tulee olla tarpeeksi asiantuntemusta molemmista johtaakseen tehokkaammin tiimiä. (Cohn 2010, 118-120.)

Cohnin 2010 mukaan Scrum Masterin ei tulisi toimia samanaikaisesti esimerkiksi kehittäjänä, testaajana tai muussa roolissa vaan mieluummin siten, että Scrum Masterin resurssia jaettaisiin yhden tai useamman tiiminvetäjäksi (Cohn 2010, 124).

### **Kehitystiimi (Development team)**

Kehitystiimin optimaalinen koko on seitsemän henkilöä. Päivittäiset päiväpalaverit muodostuvat hankaliksi, mikäli tiimin koko on liian iso. Liian pieni tiimi taas vähentää vuorovaikutuksen määrää, mikä vähentää tuottavuutta. Tiimin pitäisi koostua henkilöistä, joilla on tarvittavat taidot saavuttaakseen Sprintin tavoitteet. Scrum välttää hierarkkisia pystysuoria tiimejä, jossa analyytikot, suunnittelijat, laatutarkkailijat antavat alaspäin käskyjä kehittäjille. Tiimit ovat itseohjautuvia ja vastuullisia ja jokainen

tiimin jäsen soveltaa omaa asiantuntemustaan ongelmien ilmaantuessa. Tiimille annetaan täysi valta tehdä, mitä ikinä tarvitaan ja on välttämätöntä tavoitteen saavuttamiseksi. (Schwaber & Beedle 2002, 33-35.)

Vain tuotteen työlistan (Product Backlog) sisältö ja työmäärä ohjaavat tiimiä ja vain kehitystiimi voi päättää, mitä voidaan saavuttaa muutaman päivän sisällä. Tiimillä on päätösvalta tehdä päätökset, tehdä mitä ikinä tarvitsee ja pyytää poistamaan esteet. Scrumissa vain tiimi voi määrittää oman työnsä. Tiimin jäsenet voivat pyytää konsultointiapua, jonka saadessaan, se voi ottaa sen vastaan tai hylätä saamansa neuvon. (Schwaber & Beedle 2002, 38.)

Scrumin mukaan tiimityö perustuu viiteen arvoon, joita jokainen tiimin jäsen soveltaa ohjaamaan päätöksentekoa.

- Sitoutuminen. Ole osa tiimiä ja sitoudu tavoitteisiin, työskentele yhdessä täyttääksesi sitoumuksen.
- Fokus. Keskity tavoitteisiin. Keskity muutama asiaan kerralla. Selkeät roolit ja tavoitteet auttavat tavoitteen saavuttamisessa.
- Avoimuus. Kaikki omassa projektissa ja myös muiden projekteissa on läpinäkyvää ja avointa tarkastuksia ja parannuksia varten. Yllätyksiä ei tule enää muutaman kuukauden jälkeen. Scrumin perustana toimivat tukipilarit - läpinäkyvyys, tarkastus ja sopeutuminen.
- Kunnioitus. Jokainen tiiminjäsen valitaan omien vahvuuksien takia mukaan projektiin. Vahvuuksien mukana tulee myös heikkouksia, joiden myötä annetaan mahdollisuus oppia ja kasvaa. Jos huomaat, että joku tiimin jäsen ei ole perillä jostakin asiasta, sinun tulee auttaa häntä, koska olet vastuussa koko tiimin menestyksestä. Jokaisen tulee kunnioittaa muita.
- Rohkeus. Scrumissa on aina kyse muutoksesta, joka tulee hyväksyä. Sano hyvästit työtavoille, joita on tehty vain siitä syystä, että ”näin on aina tehty” eli rutiineja voidaan rikkoa. Muutokset vaativat aina rohkeutta. (Layton 2015, luku 1.)

## Liiketoiminta asiakkaan edustajana

Tässä yhteydessä asiakkaalla tarkoitetaan liiketoiminnan edustajaa, joka toimii asiakastarpeiden määrittelijänä ja edustaa asiakasta ohjelmistokehityksessä. Asiakas on tyytyväinen, kun hänelle tehdään tuotteita ja palveluita oikeaan aikaan. Ketterässä kehittämisessä tehdään aina ensin tärkeimmät asiat eli ne, joilla on suurin arvo asiakkaalle. Asiakas tulee ottaa huomioon kokonaisvaltaisesti ja on tärkeää ymmärtää asiakkaita ja heidän tarpeensa. Oikeat asiakkaat pitääkin saada aikaisessa vaiheessa mukaan työhön, jolloin myös palautteen saanti mahdollistuu. (Auer ym. 2013, 27.)

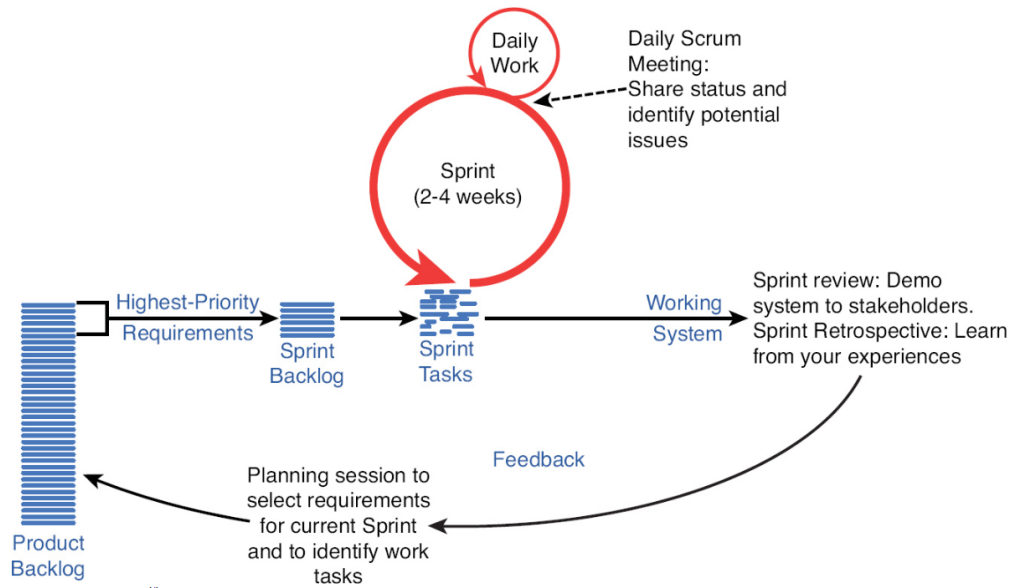
Laytonin 2015 mukaan Scrum mahdollistaa korkeampaa asiakastyytyväisyyttä, koska Scrum-tiimi on omistautunut tuottamaan tuotteita ja palveluita tyydyttääkseen asiakkaan tarpeet. Asiakastyytyväisyyttä tuovat seuraavat asiat:

- Scrum-tiimin, asiakkaiden ja sidosryhmien välinen tiivis yhteistyö heidän pitäminen mukana hankkeessa.
- Mukana on tuoteomistaja, joka on asiantuntija vaatimusmäärittelyjen ja asiakastarpeiden tunnistamisessa
- Ajantasainen tuotteen työlista on priorisoitu ja se vastaa nopeasti muutoksiin.
- Toiminnallisuuksien tuominen nopeammin ja useammin jokaisella julkaisulla mieluummin kuin kaikki kerralla ihan lopussa. (Layton 2015.)

## 5.2 Scrumin tapahtumat

Jokainen Scrumin tapahtuma on aikarajattu eli jokaiselle tapahtumalle on määritelty maksimipituus, jotta prosessissa ei pääse syntymään hukkaa. Tapahtumat ovat ennalta sovittuja ja aikarajattuja, mikä luo säännöllisyyttä. Tapahtumat on tarkasti suunniteltu lisäämään läpinäkyvyyttä ja mahdollistamaan tarkastelu, mikä on erittäin tärkeää tuotekehitykselle. Tarkoituksena on myös minimoida muiden palaverien tarve, jotta aikaa jää tuottavalle työlle. (Schwaber & Sutherland 2017, 21.) Seuraavalla sivulla kuviossa 3 esitetään, miten Scrum-prosessi etenee ja mitä tapahtumia prosessiin sisältyy.





Kuvio 3: Scrum prosessi ja aktiviteetit (Ambler & Lines 2012)

### Tuotteen työlistan (Product Backlog) muodostaminen

Kun projektin visio on selvillä eli miksi projektia tehdään ja mitä sillä halutaan saavuttaa, muodostetaan alustava tuotteen työlista tuotteeseen tarvittavista ominaisuuksista. Tuotteen työlista on lähde kaikille toteutettaville vaatimuksille ja muutoksille, mitä tuotteeseen tarvitaan. Se koostuu yksittäisistä selkeistä vaatimuksista, mitä järjestelmän on tuettava ja näillä tarkoitetaan käyttäjätarinoita (User stories). Tuotteen työlistan priorisoi tuoteomistaja ja se tulee tehdä ennen sprintin suunnittelukousta. (Ketteryys haltuun: Yleisimmät ketterät käytännöt.)

Käyttäjätarina on lyhyt ja yksinkertainen kuvaus ominaisuudesta sen tahon tai henkilön näkökulmasta, joka ominaisuutta tarvitsee. Tämä henkilö on tavallisesti joko asiakas tai ohjelmiston käyttäjä. Käyttäjätarina noudattaa usein samanlaista muotoa: **As a <type of user>, I want < some goal> so that <some reason>**. Käyttäjätarinasta tulee käydä ilmi käyttäjätyyppi, jolle ominaisuutta kehitetään, mitä ominaisuudelta halu-

taan sekä ominaisuudelta tavoiteltava arvo tai hyöty. (Cohn 2009, 239.) Käyttäjätarinan on yksi toiminnallinen vaatimus, mitä ohjelmiston tulee tukea. Käyttäjätarinasta tulee käydä ilmi, kuka ominaisuudesta hyötyy, mitä ominaisuudelta toivotaan ja miksi ominaisuutta halutaan. Käyttäjätarinan ei ole tarkoitus olla vielä tarkka määrittely, vaan muistuttaa asioista, joista pitää keskustella ja vaatimukset tarkentuvat keskustelujen yhteydessä projektin kuluessa. (Ketteryys haltuun: Yleisimmät ketterät käytännöt.)

Käyttäjätarinan tulee täyttää kolmen "C":n kriteerit: "Card, Conversation ja Confirmation". "Card" muistuttaa määrittelijää muistamaan, että käyttäjätarinan tulee mah-  
tua yhdelle pienelle paperipalalle ja siinä tulee kuvata oleelliset asiat "who", "what" ja "why". "Conversation" tarkoittaa, että käyttäjätarinan on tarkoitus olla kimmoke alkavalle keskustelulle asiakkaan, tiimin ja muiden sidosryhmien kanssa sen sijaan, että tarinaan tehtäisiin tarkka yksityiskohtainen määrittely. "Confirmation" sisältää hyväksymiskriteerit (acceptance criteria) sekä testauksen, jotka auttavat arvioimaan vastaako kehitetty toiminto käyttäjien ja asiakkaiden tarpeita ja odotuksia. (Agile Requirements Development 2018.) Alla olevassa kuviossa 4 on esimerkki käyttäjätarinasta.

**As a < role> I want < a feature> so that<reason, value**



**Who**



**What**



**Why**

**As a Client I want to upload invoice copy via system so that I can send it to my customer when needed.**

Kuvio 4: Esimerkki käyttäjätarinasta

## **Sprintin suunnittelu (Sprint Planning)**

Sprintin suunnittelukokous on yhden päivän pituinen työpaja, joka on jaettu kahteen osaan. Ensimmäisellä puoliskolla valitaan tuotelistasta seuraavan sprintin vaatimukset ja toisella puoliskolla aika käytetään seuraavan sprintin valmisteluun. Suunnittelukokoukseen osallistuvat Scrum Master, tuoteomistaja ja kehitystiimi. Tiimi päättää, mitkä osat tuotteen työlistalta valitaan seuraavaan sprinttiin ja vastaa siitä, miten sprintin työlistaksi valitut vaatimukset muutetaan osaksi julkaistavissa olevaa tuotetta. (Ketteryys haltuun: Yleisimmät ketterät käytännöt). Sprintin suunnittelupalaverin tarkoituksena on vastata kysymyksiin: mitä sprintissä tehdään ja miten valittu työ toteutetaan (Schwaber & Sutherland 2017, 13).

## **Sprintti (Sprint)**

Scrumin ydin on enintään kuukauden mittainen sprintti eli kehitysjakso, jonka sisällä tuotetaan valmiin määritelmän täyttävä, käyttökelpoinen ja potentiaalisesti julkaisukelpoinen inkrementti ohjelmistoon (Schwaber & Sutherland 2017, 21). Myös Layton (2015) korostaa, että valmiin määritelmän täyttävä ominaisuus tulee täyttää ominaisuudelle asetetut laatuksiteerit. Kehitetty ominaisuus täyttää valmiin määritelmän, kun uusi ominaisuus on kehitystiimin toimesta

- kehitetty kokonaisuudessaan
- testattu ja varmistettu, että se toimii varsinaisessa ympäristössä ilman häiriöitä
- integroitu kokonaan ohjelmistoon ja muihin sidoksissa oleviin järjestelmiin
- dokumentoitu (Layton 2015, luku 4).

Sprintin aikana voidaan toteuttaa yksi tai useampi julkaisu, joista inkrementti koostuu. Koko kehityksen ajan Sprintit ovat samanpituisia ja uusi sprintti alkaa heti toisen

päätyttyä. (Schwaber & Sutherland 2017, 9.) Myös Rising ja Norman (2000, 30) toteavat, että jokainen sprintti on aikarajattu kehitysjakso, mikä tarkoittaa, ettei sprintin päättymispäivä muutu. Syy miksi sprintin kestoa ei suositella muutettavan vaan pidettävän säännöllisenä ja aikarajattuna, on katsottu olevan hyötyä kehitystiimille, jotta se pystyy asettumaan säännölliseen työrytmiin eikä epäselvyyttä synny siitä, että ”oliko tämä viimeinen viikko ennen julkaisua” tai ”julkaisemmeko jotain tämän vai ensi viikon torstaina” (Cohn 2010, 276).

Tarkoitus on, että jokaisen kehitysjakson eli sprintin tuotoksena saadaan toimitettua jotain, jolla on välitöntä arvoa käyttäjille tai asiakkaalle, jonka he pystyvät myös näkemään omin silmin. Hyödyllistä työskentelemisestä sprinteissä on mahdollisuus saada palautetta käyttäjiltä ja asiakkaista jokaisen sprintin jälkeen ja tämä onnistuu, kun jokaisen sprintin jälkeen on jotain ominaisuuksia tuotu ohjelmistoon. (Cohn 2010, 263.)

Jos kehitystiimi toteaa, ettei kykene suoriutumaan sprintistä, se voi neuvotella tuoteomistajan kanssa, mitkä ominaisuudet voidaan siirtää seuraavaan sprinttiin. Vastavasti, jos tiimi toteaa, että olisi mahdollista toteuttaa enemmän, se voi neuvotella tuoteomistajan kanssa, mitä ominaisuuksia voidaan lisätä seuraavan sprinttiin. Tiimillä on kaksi hallinnollista velvollisuutta sprintin aikana: heidän tulee osallistua päiväpalaveriin ja pidettävä sprintin työlista ajan tasalla. Sprintin työlista on julkisesti nähtävillä ja sen kautta on mahdollista saada selville arviot jäljellä olevasta työmäärästä. (Ketteryys haltuun: Yleisimmät ketterät käytännöt.)

### **Sprintin katselmointi (Sprint review)**

Sprint katselmointiin osallistuvat Scrum-tiimi ja tuoteomistajan kutsumat sidosryhmien tärkeimmät edustajat (Schwaber & Sutherland 2017, 13). Sprintin katselmointin tarkoituksena on esitellä tuoteomistajalle ja muille sidosryhmille tuloksia, mitä sprintin aikana on saatu valmiiksi. Dokumentteja yms. ei saa esitellä eikä myöskään keskeneräisiä tuotoksia. Kehitystiimin jäsenet esittelevät toteutettuja toiminnollisuuksia ja vastaavat tuoteomistajan ja sidosryhmien kysymyksiin. Scrum Master arvioi katselmointiin osallistuvien henkilöiden määrän ja varmistaa, että katselmointi

viedään läpi asiaankuuluvalla tavalla (Ketteryys haltuun: Yleisimmät ketterät käytännöt.)

### **Sprintin jälkitarkastelu (Sprint retrospective)**

Sprintin retrospektiivi pidetään sprintin katselmoinnin jälkeen ja ennen seuraavan sprintin suunnittelupalaveria (Schwaber & Sutherland 2017, 14). Jälkitarkasteluun osallistuu kehitystiimi, Scrum Master ja tuoteomistaja niin halutessaan. Jälkitarkastelussa jokainen osallistuja vastaa kysymyksiin: mikä edellisessä sprintissä meni hyvin, mikä meni huonosti ja mitä voitaisiin parantaa seuraavaan sprinttiin. Scrum Master kirjaa ylös ja koostaa osallistujien havainnot. Scrum Master auttaa tiimin jäseniä löytämään parempia työtapoja hyödyntää Scrumia. Muutosehdotukset, jotka saadaan aikaiseksi jälkitarkastelun tuotoksena, sijoitetaan korkealla prioriteetilla seuraavan sprintin työlialle (Ketteryys haltuun: Yleisimmät ketterät käytännöt.) Kasvattaakseen tuotteen laatua Scrum-tiimi tarkastelee ja sopeuttaa ”valmiin määritelmänsä” jälkitarkastelussa.

### **Päivittäispalaveri (Daily)**

Päivittäispalaveri osallistuu jokainen kehitystiimin jäsen ja se pidetään aina samassa paikassa jokaisena työpäivänä. Päivittäispalaverin kesto on rajattu 15 minuuttiin. Palaverin tarkoituksena on tarkastella edellisen palaverin jälkeen tehtyä työtä ja ennustaa jäljellä olevaa työtä. Päivittäispalaverissa käydään lävitse kolme asiaa:

- Mitä kukin on tehnyt edellisen palaverin jälkeen?
- Mitä aikoo tehdä ennen seuraavaa päiväpalaveria?
- Mitkä seikat estävät pääsemästä tavoitteeseen?

Jokainen tiimin jäsen kertoo omalta osaltaan em. asiat. Muuta keskustelua ei sallita. Jos joku tiimin jäsen kertoo kaipaavansa apua tehtävien suorittamiseen, voi kuka tahansa järjestää erillisen palaverin päivittäispalaverin päätyttyä. Päivittäispalaveria pidetään tärkeänä työn tarkastelun ja sopeuttamisen kannalta. (Schwaber & Sutherland 2017, 12.)

## 6 Tutkimustulokset

Tässä luvussa kuvataan tutkimuksen empiirisen osuuden eli teemahaastattelujen avulla toteutetun tutkimuksen tutkimustulokset. Tämän tutkimuksen tutkimusasetelmassa esitettiin tutkimusongelmasta johdetut kysymykset: *Millaisia kokemuksia projektinjäsenillä on vesiputousmallin mukaisesta projektinhallinnasta? Millaisia vaikutuksia projektimallin muutos Scrumiin on projektille tuonut? Mitä haasteita projektimallin muutoksen jälkeen on kohdattu?*

Tutkimustuloksia esiteltäessä tulee ottaa huomioon tutkimuksen eettisyys. Keskeinen osa tutkittavien suojaa on osallistujien tutkimustietojen käsitteleminen luottamuksellisesti ja osallistujien jääminen nimettömäksi (Tuomi & Sarajärvi 2009, 131).

Laadullisessa tutkimuksessa haastateltaviksi valitaan tapaukset, joilta saa ilmiön kannalta parasta tietoa ja usein myös suositellaan aineistonlähteenä asiantuntijoiden käyttöä (Kananen 2008, 76). Tutkimuksessa haastateltiin neljää projektiryhmän jäsentä, joiden tehtäviin projektissa kuuluu vaatimusmäärittysten kirjoittaminen liiketoiminnan, asiakkaiden ja loppukäyttäjien tarpeiden perusteella kehitettävään ohjelmistoon sekä he tekevät loppukäyttäjätestausta kehitetyistä ominaisuuksista ennen tuotosten julkaisua ja käyttöönottoa. He toimivat liiketoiminnan ja asiakkaan edustajina ohjelmistokehityksessä ja tekevät tiivistä yhteistyötä ohjelmiston tuottajan kanssa. Kaikki haastateltavat ovat toimineet projektissa noin kolmen vuoden ajan, joten heillä on kokemusta ja riittävä asiantuntemus molemmista projektimalleista. Ilmiön kanssa tekemisessä olevien henkilöiden määrä oli niin pieni, että heidät kaikki voitiin valita mukaan tutkimukseen (Kananen 2015, 146).

Teemahaastatteluiden kysymykset muodostettiin kirjallisuuskatsauksen aihepiirien pohjalta. Seuraavalla sivulla on tehty tiivistämällä kuvioon 5 vertailua projektimallien vesiputous ja Scrum välillä pohjautuen haastatteluissa esiin nousseisiin asioihin.

	VESIPUTOUS	SCRUM
LUONNE	Kontrolloiva, ennalta määritelty, vaiheittainen, hidas, selkeä	Empiirinen, sopeutuva
ROOLIT	Hierarkkiset	Ei hierarkioita
VAIHEET	Alkaa tarkalla suunnittelulla, jossa kaikki vaatimukset tiedossa, päättyy kaikkien vaatimusten julkaisuun kerralla	Alkaa tavoitteista ja joistakin vaatimuksista, päättyy vaatimusten julkaisuun useissa lyhyemmissä kehitysjaksoissa
LIIKETOIMINNAN MÄÄRITTELYT	Isot kokonaisuudet, jossa kaikki liiketoiminnan tarpeet huomioitu	Käyttäjätarinat, liiketoiminnan tarpeet on palasteltu osiin
DOKUMENTOINTI	Kattava dokumentointi	Product backlog, Sprint backlog
AIKATAULUARVIOT	"Mutu"-tuntumalla, isojen kokonaisuuksien arviointi hankalaa, usein pielessä	Pienemmät osaset helpompi arvioida, enemmän paikkansapitäviä
KOMMUNIKOINTI	Liiketoiminta -> Analytit <-> kehittäjät	Liiketoiminta <-> koko Scrum-tiimi
LÄPINÄKYVYYS	Liiketoiminnalla ei näkyvyyttä miten etenee toteutusvaiheessa	Työkalut, palaverikäytännöt ja kommunikointi maksivoimat läpinäkyvyyttä
MUUTOKSET	Ei sallittuja, vaatii paluun alkuvaiheeseen, mikä hidastaa prosessia	Sallittuja, muutoksesta uusi käyttäjätarina -> nopea toteutus
PALAVERIKÄYTÄNNÖT	Ei säännöllisyyttä, tarpeen mukaan	Säännölliset, aikarajatut palaveri
TESTAAMINEN	Julkaisujen sisällöt isoja. Testaamiselle jätetty aika vähäinen valtavien kokonaisuuksien takia	Testaukseen panostetaan enemmän, testaus pienemmissä osuksissa
LÄPIMENOAIKA	Isot kokonaisuudet useita kuukausia, sisältää kaikkien vaatimusten toteutuksen	2-4 viikon kehitysjaksot, sisältää valmiin määritelmän täyttävä julkaisukelpoisen tuoteversion
LAATU	Jokaisen vaiheen hyvä suunnittelu -> hyvä laatu. Virheiden esiintyminen jossakin prosessin vaiheessa vaikuttaa lopputuloksen laatuun. Usein ei vastannut odotuksia.	Parantunut, mutta vielä parannettavaa. Valmiin määritelmän täyttävä käyttäjätarina parantanut laatua. Retrot mahdollistavat jälkitarkastelun ja jatkuvan parantamisen
HYÖDYT	Selkeä, yksinkertainen malli ohjelmille, joiden määritykset voidaan määrittellä tarkasti etukäteen eikä muutoksia matkan varrella ole odotettavissa	Yksinkertainen, hyvin omaksuttavissa oleva viitekehys monimutkaisten tuotteiden kehittämiseen, toimittamiseen ja ylläpitoon

Kuvio 5: Tutkimustulokset Vesiputous vs. Scrum

Aineistosta tehdyn analyysin perusteella aihealueet on jaettu teemoittain viiteen osaan. Tarkoituksena on kuvata, minkälaisia kokemuksia projektiryhmän jäsenillä on aikaisemmasta projektimenetelmästä vesiputousmallista ja mitä vaikutuksia Scrumin käyttöönotto projektimenetelmänä on tuonut läpinäkyvyyden, laadun, läpimenoajan ja itseohjautuvuuden suhteen. Omana otsikkona on esitetty myös teemahaastattelujen perusteella nousseet haasteet Scrumiin siirtymisen jälkeen.

## 6.1 Projektijäsenten kokemukset läpinäkyvyydestä vesiputous vs. Scrum

Haastatteluiden perusteella voidaan todeta, että kun projektia on hallittu vesiputousmenetelmällä, ei projektiryhmällä ollut näkyvyyttä kehitykseen juurikaan sen jälkeen, kun asiakastarpeet ja liiketoiminnan määritykset oli käyty läpi ohjelmiston tuottajan kanssa. Kun keskusteluja on käyty esisuunnitteluvaiheessa asiakkaan tarpeista ja lii-

ketoiminnan määrittämisestä ja ohjelmistolta vaadittavista ominaisuuksista, keskusteluissa ovat olleet mukana projektiryhmän jäseniä liiketoiminnan edustajina ja järjestelmän toimittajan taholta projektipäällikkö, Business Analyst, System Analyst ja System Architect ilman, että kehittäjät, jotka varsinaisen ohjelmistokoodin ohjelmistoon kehittävät, olisivat olleet keskusteluissa mukana. Keskusteluiden jälkeen yleensä Business Analyst otti liiketoiminnan määrittämiset analysoitavaksi ja hän alkoi käydä määrittämiä lävitse kehittäjien kanssa omassa päädyssään. Jos asioita tarvitsi tarkentaa liiketoiminnasta tai keskustella vielä lisää, erinäisiä palaveriteita kutsuttiin koolle niin kauan kuin ”analyst” koki saavansa täyden ymmärryksen vaatimuksista.

*liiketoiminnassa me kommunikoidaan analyystille jollain tavalla ja hän sitten tavallaan käy sen loppukeskustelun siellä. Hän ja hänen näkemykset siitä, ja sitten ne kehittäjät. Että sitten meiltä liiketoiminnasta katkeaa se näkymä siihen, et mitä oikeastaan sen jälkeen tapahtuu, kun me ollaan ikään kuin kerrottu se asia. Oikeastaan seuraava näkymä on sitten siinä, kun se tulee taas hyväksymisestä kasaan, kun se on jo valmis*

*Me ei sinänsä päästy suoraan olemaan juuri ollenkaan yhtään niiden koodareiden kanssa tekemisissä. Meillä oli niin sanottu filteri siinä välissä, joka sitten käänsi meidän määrittämiset vähän erinäköiseen muotoon, ja vasta sitten annettiin koodareille.*

*Siinä vanhassa oli se keskustelu niiden analystien kanssa, että niiden kanssa me keskusteltiin, ja ne kertoi sitä eteenpäin. Mikä on ehkä vähän semmoinen kun on kuitenkin kielimuuri tietyllä tavalla välissä.*

Scrum-malliin siirtymisen myötä kommunikaatio on parantunut huomattavasti liiketoiminnan ja ohjelmistotuottajan välillä ja tämä on yksi merkittävä tekijä siihen, että myös läpinäkyvyys on parantunut verrattuna aikaisempaan malliin. Scrum-mallissa kommunikoidaan suoraan liiketoiminnan edustajien ja kehittäjien välillä, kun aikaisemmassa mallissa oli ”analyst” välissä, jolle kommunikoitiin liiketoiminnan tarpeista ja joka kommunikoi kehittäjien kanssa ja toimi ikään kuin tiedonvälittäjänä välissä.



Vesiputousmallin aikaan palavereja pidettiin, mutta ne eivät olleet säännöllisiä vaan niitä pidettiin tarpeen mukaan.

*ei ole oikeastaan ollut mitään säännöllisiä palavereja siinä vaiheessa. Että se on sitten tarpeen mukaan, että jos on jotain asioita, mitä pitää käydä läpi siihen liiketoiminnan määrittelyyn liittyen, niin sitten on varattu. Mutta ei ole ollut mitään säännöllistä niiden kehittäjien kanssa.*

Haastateltavista suurin osa oli sitä mieltä, että myös Scrumin myötä tulleet aikarajatut ja säännölliset palaverikäytännöt ovat hyviä ja tarpeellisia ja ne omalta osaltaan lisäävät läpinäkyvyyttä. Etenkin sprintin katsauspalaverit, joissa kehittäjät esittelevät omia tuotoksiaan, koettiin erityisen hyödyllisinä, koska tilaisuuden aikana liiketoiminnan edustajat ja muut sidosryhmät voivat omin silmin nähdä, mitä on saatu edistettyä ja voivat esittää kysymyksiä kehittäjille näin halutessaan. Kaikki Scrumiin liittyvät aikarajatut palaverit on koettu hyödylliseksi, erityisesti palaverit, joissa myös projektiryhmän edustus on paikalla.

*Suomen päästä meillä on osallistujat mukana aina näissä Sprint Reviewissa eli näissä demosessioissa, mitkä on tullut myös tämän uuden mallin myötä säännöllisinä palaverina ja on todella hyviä. Todella hyvä, että näitä pidetään. Toisaalta myös retrot säännöllisesti, jossa me ollaan mukana, ne on erittäin hyviä, että säännöllisesti käydään läpi se edellisen sprintin onnistuminen ja ehkä epäonnistumiset, ja koitetaan mieltä korjaavia toimenpiteitä aina tällä nopealla syklillä. Myös tämä Sprint Planning on semmoinen, missä on Product Owner mukana. Elikä pääsee juurikin sanomaan, että mitä sinne seuraavaan sprinttiin aina kulloinkin otetaan mukaan. Kaikki nämä on erittäin tarpeellisia ja hyviä olleet.*

*Ne Sprint Review't on ollut hirveän hyviä, että ne kertoo sen mitä he ovat tehneet, tai näyttää miten se toiminnallisuus.. Koska sitä on paljon*

*helpompi sitten mennä lukemaan se, että mitä on haluttu ja sitten testata se meidän kannalta. Mä olen kyllä tykännyt niistä*

Yksi haastateltavista puolestaan totesi, että palavereihin eivät osallistu kaikki projektin jäsenet ja tästä syystä palaverikäytännöt eivät Scrumin myötä ole merkittävästi muuttuneet.

*No eihän meillä nytkään kauheasti palaverikäytäntöjä loppujen lopuksi ole. Että meillä on demot tai – nimellä. Mitkä meillä tietyssä mielessä oli aikaisemminkin, ja sitten Retro-palaveri jossa on vaan minä ja Product Owner meidän puolelta. No se on ehkä jotain lisää mikä tavallaan on hyvä, mutta ei meillä edelleenkään ole mitään säännöllisiä palavereita kehittäjien kanssa. Tai anteeksi refinementit on toki sitten tuolla, mutta jos ajatellaan sellaista yleistä, että mistä nyt keskusteltaisiin niin ei ole. Että se perustuu edelleenkin sitten siihen tarvittaessa.*

Haastatteluista nousi esiin, kun liiketoiminnan määrittelyt oli toimitettu ohjelmiston tuottajalle, ei oikein kenelläkään projektiryhmän jäsenellä ollut näkyvyyttä kehitykseen moneen kuukauteen. Tuoteomistajan rooli on tärkeä rooli erityisesti läpinäkyvyyden kannalta ja roolin selkeytyksen kautta ongelma on korjaantunut. ”Meillä on valta mihin halutaan kehitysresursseja käyttää ja työkalut seurata etenemistä.”

Haastatteluiden mukaan koettiin tärkeäksi, että tuoteomistajan rooliin nimetty henkilö istuu nimenomaan projektiryhmässä liiketoiminnan puolella, jossa parhaiten osataan tunnistaa ja määritellä asiakastarpeet ja sen mukaisesti voidaan priorisoida ominaisuuksia ohjelmistoon.

Vesiputousmallissa liiketoiminnan vaatimusmäärittelyt on jätetty Business Analystin kautta kehittäjille vapaisiin käsiin ja he ovat saaneet kehittää ohjelmistoa, mihin suuntaan ovat halunneet ja projektiryhmä ei nähnyt mitään tuloksia ennen testausta, koska ohjelmistoa ei ole nähty välissä. Tällä hetkellä Scrumin myötä kuka tahansa voi mennä katsomaan, mitä tällä hetkellä menossa olevassa sprintissä on sisällä, missä

vaiheessa kehitys on menossa ja kuinka paljon ominaisuuksista on jo valmistunut, kun käytössä on oikeanlaiset työkalut, jonne jokaisella projektinjäsenellä on pääsy.

Scrumin mukainen sprintti-tyyppinen malli, jossa kehitystä tuodaan ohjelmistoon pienemmissä osissa ja kehitystä tapahtuu pikkuhiljaa, on osoittautunut myös toimivaksi malliksi läpinäkyvyyden ja ominaisuuksien oikea-aikaisen toimituksen suhteen.

*..sprintti -mallinen kehitys. Ensinnäkin me päästään vaikuttamaan siihen sisältöön, mitä siellä kaks viikkoisessa sprintissä tehdään. Me koko ajan nähdään, miten se edistyy. Sprintin loppuessa viimeistään nähdään, että mitä tuli valmiiksi ja mitä mahdollisesti jäi kesken. Kyllähän tämä on tosi nopea prosessi ja nähdään joka vaiheessa, että mitä tapahtuu ja miten on mennyt*

## 6.2 Projektijäsenten kokemukset tuotosten laadusta vesiputous vs. Scrum

Teemahaastattelussa tiedusteltiin haastateltavilta tuotosten laatuun liittyvistä tekijöistä sekä vesiputousmallin että nyt ketterän Scrumin aikana. Kuviossa 6 on tehty vertailua projektimallien välillä ja laadittu haastatteluista esiin tulleiden tekijöiden vaikutusta laadun toteutumiseen.

	VESIPUTOUS	SCRUM
<b>LIIKETOIMINNAN MÄÄRITTELYT</b>	Laajat dokumentit, jossa kaikki liiketoiminnan tarpeet huomioitu	Käyttäjätarinat, liiketoiminnan tarpeet on palasteltu osiin
<b>TESTAAMINEN</b>	Julkaisujen sisällöt isoja. Testaamiselle jätetty aika vähäinen valtavien kokonaisuuksien takia	Testaukseen panostetaan enemmän, testaus pienemmissä osuuksissa. Toimivat työkalut – selkeämpää mitä testataan
<b>LAADUN TARKKAILU</b>	Jokaisen vaiheen hyvä suunnittelu -> hyvä laatu. Virheiden esiintyminen jossakin prosessin vaiheessa vaikutti lopputuloksen laatuun. Virheet ilmi loppuvaiheessa. Laatu usein ei vastannut odotuksia.	Parantunut, mutta vielä parannettavaa. Valmiin määritelmän täyttävä käyttäjätarina parantanut laatua. Retrot mahdollistavat jälkিতarkastelun ja jatkuvan parantamisen Testaukseen panostetaan enemmän, testaus pienemmissä osuuksissa

Kuvio 6: Laatuun vaikuttavat tekijät vesiputous vs. Scrum

Ohjelmistoon kehitettyjen tuotosten laatu ei vesiputousmallin aikaan vastannut aina liiketoiminnan odotuksia. Laatuun on vaikuttanut liiketoiminnan määritysten laatiminen isoissa kokonaisuuksissa, joihin tuli miettiä kaikki asiakkaan tarpeet tarkasti ja kirjoittaa sanatarkasti, kaikki mitä asiaan liittyi, missä määritykset ja tarpeet tulevat näkymään ja mihin ne vaikuttavat. Vaatimusmäärittelyjä kirjoitettiin dokumenteille, jotka saattoivat muodostua monisivuisiksi. Niiden laatiminen koettiin monimutkaisena ja kankeana prosessina ja haasteena oli myös vastapuolen ymmärryksen saaminen järjestelmältä odotetuista vaatimuksista.

*mitä me kutsuttiin määrityksiksi, oli tämmöisiä aivan massiivisia dokumentteja tai kirjoituksia siitä, että mitä tämmöinen yleensä vähän isompi kokonaisuus, joku isompi toiminnallisuus tai isompi asia, mitä me tarvitaan, niin mitä kaikkea se sisältää.*

*Sun on aina tosi vaikea kirjoittaa kirjallisesti jotain niin sanatarkasti, kun aina sinne sisältyy joku oletamus. Että kun sä omassa mielessä näet sen, niin se, että sä saat sen kaiken siihen paperille, niin on lähes mahdotonta. Eli vaikka suoritukset, niin kaikki mahdollinen, mitä sinne alle liittyy niin se oli niin kuin yksi osio. Mietittiin nyt että vaikka tietty meidän asiakas, että mitä se nyt tästä suoritus osa-alueelta just tarvitsee, ja niitä tehtiin sitten. Mutta että ne saattoi siltikin suurimmillaan olla erittäin monisivuisia dokkareita*

*Mutta sitten vastapuolella varmaan tulee vastaan se, että jos sä oot koodari/kehittäjä, niin et sä jaksa lukea mitään kymmenen tai ylikin sivusta dokumenttia, ennen kuin sä alat tekemään, vaan se pitäisi olla aika yksinkertaista – ja sitten siinä tietysti käy niin, että ne on tehneet jotain sinnepäin...varmasti ei saatu, sellaista mitä me lähtökohtaisesti ollaan lähdetty hakemaan. Ainahan releasien jälkeen on ollut jonkinlaisia incidentteja ja asioita on mennyt rikki, sellaisiakin jota on kertaalleen korjattu.*

Haastatteluissa todettiin, että isompia ja pienempiä virheitä järjestelmästä löytyi jonkin julkaisujen jälkeen, mikä aiheutti tyytymättömyyttä projektissa ja asetti kysymyksiä myös muiden sidosryhmien taholta. Sen lisäksi, että laatua heikensi isojen kokonaisuuksien hallinta ja määritykset, jossa kaikkea ei oltu osattu ottaa huomioon, on lopputuotosten laatuun vaikuttanut myös testauksen ajan vähyys. Koska valtaosa testauksesta tehtiin projektin loppuvaiheessa ja testattavien kokonaisuuksien ollessa niin isoja, ominaisuuksia ei ehditty testaamaan kunnolla.

*Siinä vaiheessa kun meillä oli jo testaus käynnissä, että näissä määrittelyissä olisi voinut, niitä ei välttämättä oltu tehty tarkasti tai sitten ei oltu osattu miettiä kaikelta kannalta sitä*

*meillä ei ollut riittävästi liiketoiminnan päässä aikaa testata niitä asioita. Releasien sisällöt oli niin massiivisia, että siihen testaukseen olisi pitänyt varata paljon enemmän aikaa, ja kaiken lisäksi vielä yleensä vielä kutistu hyvin minimiin, että se testaaminen oli haastavaa hallita myöskin tässä, kun kokonaisuudet oli niin isoja*

Suurin osa haastateltavista oli sitä mieltä, että laatu Scrum-menetelmän kautta on parantunut, mutta vielä ei voitu todeta, että se on erinomaista. Tiivis yhteistyö ja aktiivinen kommunikointi liiketoiminnan ja kehitystiimin välillä on läpinäkyvyyden parantumisen lisäksi nostanut myös tuotosten laadun tasoa. Kommunikoinnin myötä asioita tulee esille jo kehitysvaiheessa eikä vasta lopussa, kuten vesiputousmallin aikaan.

*Selkeästi parantunut siitä ehkä just sen kommunikoinnin myötä, että tulee niitä asioita usein esille jo siinä kehitysvaiheessa, että ei ole aina vasta siellä lopussa.*

Yksi merkittävä tekijä, mikä nousi haastatteluissa esiin laadun parantumiseen vaikuttavana tekijänä Scrum-malliin siirtymisen myötä, oli vaatimusmääritysten kirjoittaminen pienemmissä osissa käyttäjätarinoiden muotoon sen sijaan, että määrityksistä laadittaisiin laajat tarkat dokumentit, jossa kaikki asiat tuli ottaa kerralla huomioon.

Vaatimusmäärittysten pilkkominen pienempiin osiin helpottaa sekä liiketoiminnan määrittelijän työtä ohjelmistolta odotettavista ominaisuuksista että vastapuolen ymmärryksen saamista asiasta.

*Ei ole semmoista massiivista, mihin pitäisi kaikki yhtäkkiä miettiä, vaan mieltii sen yhden pienen kokonaisuuden kerrallaan. Se on paljon helpompaa*

*Kyllähän se nyt on paljon yksityiskohtaisempaa, että me tiedetään ihan tikettitasolla että mitä siihen storyyn laitetaan ja valta on meidän käsissä. Vois olla helpompi miettiä, että kun sä otat vaan yhden asian niin sitten sää nyt mietit vaan sen, kun sen kerralla kuntoon. Sulla ei mene siihen kuukautta, vaan sä saat sen ehkä tunnissa tehtyä. Kehittäjän puolelta varmasti pystyy ottaa kopin sellaisesta pienemmästä kokonaisuudesta ja siten me pystytään myös niin kuin testaamaan ja validoimaan se, kun se on vaan semmoinen yks pieni juttu sen sijaan että me validoidaan jotain hirveätä kokonaisuutta.*

Jokaisen käyttäjätarinan tulee täyttää laatukriteerit eli ”valmiin määritelmä”, mikä vaikuttaa myös laadun parantumiseen. Kun valmiin määritelmän kriteerit on muodostettu, asettaa se kehitystiimille selkeän tavoitteen, mitä pitää saada aikaan, että voidaan toteuttaa julkaisukelpoinen tuote. Keskenpästä tai virheellistä tuotosta ei päästetä julkaisuun. Valmiin määritelmä sisältää myös testauksen ennen kuin ominaisuus tulee projektiryhmän testattavaksi. Testaukseen panostetaan siis enemmän. Testaamista auttaa myös pienemmät osiot, jolloin ei kerry massiivista määrää testattavaa.

*He panostaa testaukseen enemmän. On määritelty nämä definition of donet, ja tällaiset kriteerit, että jokaisen käyttäjätarinan pitää täyttää ennen kuin se on valmis. Nyt se sisältää myös nämä testaukset ja muut tällaiset. Sekin jo itsessään parantaa sitä laatua, et ennen kuin me päästään testaamaan asioita, niin ne on varmuudella testattu. Ja bugitkin ja kaikki korjattu.*

*Ei ole niin semmoista, että oikeastaan saisi hylätä kaiken ensimmäisellä kierroksella ja sitten tehdä sen uudestaan...se että mitä meillä tulee testaukseen, niin se toimivuus on huomattavasti parempi kuin aikaisemmin ja se on ehkä selkeämpää nyt se, että mikä se muutos on ollut ja mitä mun täytyy testata.*

*Että jos mä nyt peilaan sitä että miltä se nyt toistaiseksi on näyttänyt, niin mun "mutu" on se, että se laatu on parempaa kuin aiemmin. Ei se vieläkkään virheetöntä ole*

Haastateltavien mielestä Scrum-menetelmään siirtyminen oli tuonut selkeyttä ja tehokkuutta testaamiseen, koska testattavien ominaisuuksien tarvittavan tiedon löytämiseen oli paremmat työkalut ja testattavat kokonaisuudet olivat pienempiä, kun ei tarvinnut isoja kokonaisuuksia ja kaikkia mahdollisia ominaisuuksia testata kerralla.

Käyttäjätarinan valmiin määritelmän kriteerien asettaminen edesauttaa myös keskustelua projektiryhmän ja kehitystiimin välillä sen hetkisestä tilanteesta paremmin, mitä on jo saatu valmiiksi ja mitä tulee vielä tehdä, että voidaan julkaista laadullinen tuotos. Mikäli ominaisuus ei vastaa valmiin määritelmää, se palautetaan tuotejonoon eli huonoa tai keskeneräistä ominaisuutta ei päästetä tuotantoon.

Koko prosessin laatutarkkailua tehdään myös yhteisissä palavereissa ja pyritään korjaamaan toimintatapoja tai -prosesseja, jos epäonnistumisia on esiintynyt. Sprintin päätteeksi pidettävä retrospektiivinen palaveri nähtiin tärkeänä ja hyödyllisenä palautteen antamisen ja jatkuvan prosessin kehittämisen näkökulmasta. Palautteen avulla saadaan parannuksia työtapojen ja materiaalien kehittämiseen.

*Myös retrot säännöllisesti, jossa me ollaan mukana, ne on erittäin hyviä, että säännöllisesti käydään läpi se edellisen sprintin onnistuminen ja ehkä epäonnistumiset, ja koitetaan miettiä korjaavia toimenpiteitä aina tällä nopealla syklillä*

Tuotosten laatu on projektimallin muutoksen myötä parantunut, mutta virheitä julkaisujen jälkeen on silti vielä esiintynyt. Laadussa on siis vielä parannettavaa. Eräs haastattelijoista oli sitä mieltä, että tähän voi vaikuttaa seikka, että uudet toimintatavat ja roolit on vasta otettu käyttöön ja uutta opeteltavaa on tullut kaikille. Optimistisuutta laadun parantumiseen löytyy haastatteluiden perusteella.

### 6.3 Projektijäsenten kokemukset läpimenoajasta vesiputous vs. Scrum

Vesiputousmallissa saattoi kulua kuukausia määrittelyvaiheesta siihen, että projektissa nähtiin mitään valmista. Aikatauluarvioissa ei ollut selkeyttä ja arviot menivät usein pieleen, koska isoja kokonaisuuksia oli vaikea arvioida. Projektin aikataulussa esiintyi ei välttämättä päivien tai viikkojen vaan jopa useiden kuukausien viiveitä. ”Eihän ne koskaan tulleet ajallaan. Se venyi ja venyi vaan, niin se oli hyvin turhauttavaa”

*oli tiettyjä kategorioita; S, M, L, XL. Sitten niissä oli jotkut tuntiarviot, jota ne niin sanotusti vastaisi. Mutta sehän siinä huomattiin pitkin matkaa, et ei ne estimaatit oikeastaan kertonut mitään, ja toisekseen ne oli aikalailla pielessä. Sinänsä meillä ei ollut minkään näköistä kunnollista estimointitapaa tai -mahdollisuutta siinä mallissa olemassa.*

*oli arvioitu, että helmikuussa saadaan joku asiakasryhmä migroitua, tietyt toiminnallisuudet on siihen mennessä valmiita, niin se saattoi olla-kin, et se onkin yhtäkkiä elokuu, kun se oikeasti tapahtuu. Ne oli ihan älyttömien välillä ne viiveet.*

Haastateltavien mielestä sen lisäksi, että pienempien palasten eli käyttäjätarinoiden kirjoittaminen tarvittavista ominaisuuksista on Scrumiin siirtymisen myötä helpottunut ja nopeutunut, saadaan ne myös nopeammin kehitystiimille esiteltäväksi. Pienempiä kehityksien valmistumisaikaa on myös helpompi arvioida aikataulullisesti.

Mielipiteissä aikataulujen pitämisen suhteen oli kuitenkin hajontaa vastausten välillä. Osan mielestä pienempien kokonaisuuksien työmääriä on ollut helpompi estimoida ja arviot aikataulujen suhteen ovat paremmin kohdillaan ja paikkansapitäviä. Lisäksi



niitä on helpompi seurata miten edistyvät. Yksi haastateltavan kuitenkin totesi, etteivät sprintit kuitenkaan ole toteutuneet täysin suunnitellusti, vaan niissä on ollut viiveitä myös Scrumiin siirtymisen jälkeen. Myös Scrumin aikaan on käynyt niin, ettei sprintin aikana oltu pysytty aikataulussa ja saatu kehitettäviä ominaisuuksia valmiiksi.

*Sprintit ei ole toteutuneet ihan täysin suunnitellusti. Toisaalta siinä syy-päänä on myöskin niin kuin QA testauksen puute. Ja ehkä sellainen tietynlainen resurssipuute. Se että tarkoittaako se että arvio on ollut väärässä – ei välttämättä. Mutta toisaalta jotta se olisi niin kuin toimiva se prosessi niin tämän kaiken pitäisi soljua. Että jos me sanotaan, että toi on ton kokoinen, me lasketaan että se mahtuu sprinttiin, niin silloin se pitää siellä valmistuakin.*

Vesiputousmalli on pakottanut liiketoiminnan edustajat kirjoittamaan yksityiskohtaiset dokumentit tarkasti ja laajasti ottaen kaikki asiakastarpeet huomioon myös sellaisista kohdista, joita ei välttämättä ole ymmärretty hyvin eivätkä kaikki tarpeet ole olleet sillä hetkellä tiedossa. Ja siitä syystä, että näkyvyyttä kehityksen toteutusvaiheessa ei ole ollut, ovat lopputulokset konkretisoituneet liiketoiminnalle vasta prosessin loppuvaiheessa. Tarpeet ovat saattaneet välissä muuttua tai niihin on haluttu täydennyksiä ja viilauksia. Halutut muutokset ovat vaikuttaneet ratkaisevasti tekemisen alkupäässä tehtyihin ratkaisuihin, ettei muutoksiin ole suhtauduttu järjestelmän tuottajan taholta suopeasti eikä niille näin ollen ole annettu sijaa. Muutokset kehityksen loppuvaiheessa ovat hidastaneet prosessia merkittävästi, koska usein on jouduttu palaamaan aikaisempiin vaiheisiin ja aloittaa prosessi ikään kuin alusta.

*Huomattiin, et ei ehkä sittenkään ihan noin, vaan viilataanpas pikkasen. Mikä sitten taas tässä mallissa oli sieltä kehityksen päästä ehdoton ei, et näin ei saisi missään nimessä tehdä, että näitä muutetaan. Mutta meillä koettiin se hankalaksi, kun pakkohan meidän on muuttaa, jos me huomataan, et joku pieni kohta ei olekaan oikein*

*Ja sitten jos haluttiin jotain yhtäkkistä extraa muutosta, niin periaatteessahan se tarkoitti, että vasta niiden kuukausien jälkeen se voitiin ottaa edes mahdollisesti työn alle*

Haastateltavien tuntemusten mukaan Scrumin mallissa muutokset vaatimuksiin eivät merkittävästi hidasta prosessia kuten vesiputousmallin aikaan.

*tässä kohtaa ei tule enää sitä, että tarvitsee se koko homma tehdä uudestaan vaan enemmänkin sitä; "okei, mennään tällä ja korjataan toi pikku juttu sitten vielä myöhemmin." Tai jos se on tosi pieni, niin se ehdistään vielä tässäkin. Kyllä mä näkisin, et se on paljon nopeampaa. Kun ei tarvitse sitä koko sykliä alusta asti käydä läpi, vaan pystytään tekemään se pieni osa vaan siitä uudestaan.*

*Siinä voidaan sitten arvioida, että korjataanko se tarina, käydään tiimin kanssa se uudestaan läpi muutoksen kera, tai sitten kirjoitetaan sinne jo kokonaan uusi tarina. Eli me voidaan helposti muuttaa sitä käyttäjätarinalia, mikä meillä on siellä backlogissa, et mitä me sieltä sitten nostetaan yks kerrallaan sprintteihin.*

#### 6.4 Projektijäsenten kokemukset itseohjautuvuudesta vesiputous vs. Scrum

Haastatteluiden perusteella kokemuksia kehitystiimin itseohjautuvuudesta vesiputousmallin aikaan ei juurikaan ollut projektiryhmällä, koska yhteistyö oli hyvin vähäistä liiketoiminnan ja kehittäjien välillä. Yksi haastateltavista oli kuullut, kun vesiputousmallin aikana oli määräyksiä annettu kehittäjille koodattavaksi, heillä ei varsinaisesti ollut ollut käsitystä siitä, mitä he tekevät ja minkä vuoksi. Koska keskusteluyhteyttä ei ollut ja määräyksistä keskusteltiin välikäden kautta, oli ohjelmistolta liiketoiminnalle tavoiteltu hyöty jäänyt kehittäjille epäselväksi.

*Yhden kehittäjän suusta kuulin - hän sanoi ihan suoraan, että silloin aikaisemmin hehän vaan puuhaili jotain ja ei oikeastaan itselläkään ollut käsitystä, et mitä he tekee, ja kunhan nyt vaan tekivät jotain*

Uusi malli on muuttanut kommunikaation täysin liiketoiminnan ja kehittäjien välillä. Enää ei ole "filteriä" välissä vaan keskustellaan suoraan ihmisten kanssa, joiden kanssa kuuluukin keskustella. Tällä on iso vaikutus itseohjautuvuuteen

*Nämä kehittäjät itsekin on oma-aloitteisesti meihin yhteydessä siinä vaiheessa, kun he tekee sitä koodia ja jotain toiminnallisuutta rakentaa, ja heillä tulee joku kysymys, niin he heti kysyy sitä meiltä. Ja me päästään sitten vastaamaan siihen. Tämmöistä ei tapahtunut ollenkaan oikeastaan silloin vanhan mallin aikaan.*

Haastatteluissa tuli esiin, että lokakuussa 2018 organisaatioon valittuun Scrum Masteriin ollaan erittäin tyytyväisiä liiketoiminnan puolella. Scrum Masterilla on merkittävä rooli tiimin jäsenten itseohjautuvuuden muovautumisessa. Scrum Masterin rooli on ollut toimia fasilitaattorina ja valmentajana, joka sparraa omaa tiimiään, mutta ei välttämättä puutu siihen, mitä määritellään ja miten ominaisuutta kehitetään. Scrum Master antaa vapaa kädet kehittäjille ja kehittäjät ottavat vastuun omasta työstään eikä sitä tarvitse "ulkoistaa" enää "analystille" vaan kehittäjät oikeasti tekevät työn, mikä heille kuuluu. Jos kehittäjillä on jotain epäselvää, he voivat kysyä suoraan liiketoiminnasta asiasta eikä välissä ole suodatinta.

*Scrum master on ollut aivan loistava ja erittäin tarpeellinen, ihan elinehto tälle kaikelle, että on voinut onnistua tämä uuteen malliin siirtymisen. Siinä varmaan on myös tämän ihmisen omalla luonteella paljon merkitystä, että tämä meidän Scrum Master on ollut just oikeanlainen ihminen roolissansa.*

Haastateltavien tuntemusten mukaan hierarkiat ovat madaltuneet Scrumiin siirtymisen myötä. Scrum Master on korostanut tiimeille, että tiimeillä on yhteinen vastuu asioista. Heidän tulee toimia tiiminä ja ottaa se vastuu ja hoitaa ne asiat, mitkä he

ovat lupautuneet tekemään. Scrum Master eikä kukaan muukaan kuitenkaan toimi tiimin ylätason vahtina, että tehtävät tulee hoidettua vaan Scrum Master luottaa tiimiin ja sen itseohjautuvuuteen asioiden loppuunsaattamiseksi. Scrum Master ohjaa oikeaan suuntaan, mutta ei valvo, että työ tulee tehtyä.

## 6.5 Scrumiin siirtymisen myötä kohdatut haasteet

Tämän kappaleen tarkoitus on kuvata haastatteluissa esiin nousseita haasteita, mitä Scrum-menetelmään siirtymisen myötä on esiintynyt projektiryhmän näkökulmasta. Tarkoituksena on tuoda vastauksia tutkimusongelman kolmanteen kysymykseen

*Mitä haasteita projektimallin muutoksen jälkeen on kohdattu?*

Haastatteluista kävi ilmi, että vaikka vaatimusmäärittelyjä tehdäänkin Scrumin toimintatapojen mukaan pienemmissä osissa, riskinä nähtiin, että ”langat karkaavat kädestä”, mikäli isoa kuvaa ei ole täysin kirjoitettu ja ymmärretty. Jos lähdetään tekemään isoa kokonaisuutta, on se jollain tasolla kuvattava joka tapauksessa laajemmin, että voidaan pilkkoa se ominaisuuksiltaan pienempiin osiin. Isojen toiminnollisuuksien kokonaisuus on vesiputousmaisesti kuitenkin piirrettävä ja hahmotettava, että ominaisuutta voidaan lähteä tekemään palasina. ”Muuten voi mennä pieleen, jos lähdetään ajattelemaan vain pala kerrallaan.”

Vaikka kommunikaatio on merkittävästi lisääntynyt Scrumin myötä projektiryhmän ja järjestelmän tuottajan välillä, on ongelmia esiintynyt eri ryhmien ja projektien välisessä kommunikaatiossa organisaation sisällä.

*yhteistyö jos ja kun se ei ole ihan parasta mahdollista, niin se aiheuttaa kyllä haasteita. Siinä vois olla petrattavaa ainakin. Tässä varmaan just meidän organisaatiossa olisi petrauksen paikka siinä, miten täällä ei-kehitystiimin puolella toimitaan.*

Haastatteluissa ilmaistiin huoli myös siitä, että vaikka projektiryhmässä on aloitettu toimimaan ketterästi, ei organisaatio ole luonnostaan tarpeeksi ketterä. Ongelmia on koettu prosesseissa tehdä hitaasti päätöksiä, mikä on aiheutunut odottamisena, että oikea instanssi on saanut kokoonnuttua ja tehtyä päätöksen. Kun jostain päin tulee jonkinlainen vaatimus ja asioita tulee edistää, niin jonkinlainen kokous on kutsuttava koolle päättämään asioista, mitkä venyttävät asioiden eteenpäin viemistä. Tämä on tehnyt ajoittain työnteosta haasteellista ja on ristiriidassa ketterien periaatteiden kanssa. Haastattelut toivat esiin, että myös muulta liiketoiminnalta odotetaan ketteryyttä.

*tämä meidän organisaatio ei ole luonnostaan ketterä. Sitten kun yritetään sen sisällä tehdä ketterää, niin sitten kuitenkin jostain tulee joku vaatimus, et pitää taas joku konklaavi kokoontua päättämään jostain asioista, mikä sitten taas ei ole ollenkaan Scrumia millään tavalla. Se on semmoinen huono puoli, et se organisaatio ei taivu tähän malliin ihan täysin.*

*meidän projekti on ehkä ainut missä mä aidosti tunnistan, että me oikeasti ollaan ketteriä, tai ainakin yritetään olla. Ehkä aika monessa muussa me ollaan jossain siellä välimaastossa. Et mä en koe, että jos mä menen jonkun muun kanssa jostain muusta asiasta juttelee, et nyt me ollaan ketterässä kiinni, että nyt pitäisi mennä näin. Vaan siellä herkästi peilaantuvat sitten vanhat mallit*

## **7 Johtopäätökset**

Perinteisessä vesiputousmallissa seuraavaan vaiheeseen edetään, kun edellinen vaihe on saatu valmiiksi, eikä tämän jälkeen aikaisempaan vaiheeseen ole suotavaa enää palata, koska se hidastaa prosessia ja aikataulut saattavat pettää. Perinteinen projektimalli sopii projekteihin, joissa vaatimusmäärittely ja suunnittelu tehdään huolellisesti. Tämä edesauttaa toteutusvaiheen suoraviivaista toteuttamisesta, jos

huomattavia ongelmia ei paljastu matkan varrella. Jos ongelma havaitaan aikaisessa vaiheessa, se voidaan vielä korjata, mutta jos ongelma havaitaan myöhäisemmässä vaiheessa, tuo haasteita projektille ajallisesti, työmäärällisesti ja rahallisesti.

Haastatteluiden perusteella kritiikki vesiputousmallia kohtaan oli hyvin voimakasta. Malli nähtiin kuitenkin ainoana vaihtoehtona projektimallina, kun järjestelmää on aloitettu rakentamaan tyhjästä ja koko järjestelmä tuli mieltä isoina kokonaisuuksina. Ohjelmistoprojekti oli niin massiivinen, että olisi ollut vaikea suunnitella ja viedä projektia eteenpäin ainakaan täysin Scrum-tyyppisesti, pienissä paloissa ilman vesiputousmallin elementtejä. Ja vesiputousmallin mukaisella projektinhallinnalla projektissa on kuitenkin saatu paljon aikaan. Vesiputousmalli sai haastateltavien taholta myös positiivista palautetta sen selkeyden ja yksinkertaisuuden vuoksi. Vaikka kehitys oli hidasta ja aikataulut pettivät usein eikä läpinäkyvyyttä kehitykseen ollut, oli isot kokonaisuudet helpompi hahmottaa ja menetelmän avulla pystyi olemaan selvillä, mitä tulee ohjelmistoon missäkin vaiheessa.

Scrum sopii sellaisten ohjelmistojen kehitykseen, jossa ei ole tarvetta tietää tarkemmin tulevasta tai tulevaisuuden tarpeita on vaikea ennustaa. Scrum-menetelmä pyrkii vähentämään riskejä jakamalla ohjelmistokehitystä pienempiin osiin ja lyhyempiin kehitysjaksoihin, jotka kestävät kahdesta neljään viikkoa. Kutakin lyhyttä kehitysjaksoa voidaan pitää omana pienenä projektina, mikä sisältää suunnittelun, vaatimuserittelyt, toteutuksen, testauksen ja julkaisun. Ketterä Scrum pyrkii julkaisukelpoiseen ohjelmistoon jokaisen kehitysjakson lopussa. Isona hyötynä on, että asiakkaalle tai asiakkaan edustajille voidaan demota useammin, mitä on saatu valmiiksi ja asiakas voi arvioida onko toteutus sitä, mitä on odotettu. Tuottaminen on jatkuvaa ja ohjelmistoversioiden tulee olla sellaisia, jolla voitaisiin mennä tuotantoon. Välituotoksissa saadaan tuotettua oikeasti liiketoiminnalle ja asiakkaalle arvoa tuottavia ominaisuuksia.

### **Scrumin vaikutus läpinäkyvyyteen**

Scrumin ja ketterien menetelmien erottuva piirre perinteisestä mallista on yleisesti ottaen niiden läpinäkyvyys. Selkeiden ja helposti saatavilla olevien viestintäkanavien

kautta tietoa säteilee läpi koko organisaation. Koko organisaatio pystyy olemaan tietoinen, mitä on tehty, minkä parissa on työskennelty ja mitä on vielä tekeillä. (Layton 2015, luku 1.)

Perinteisen mallin mukaisessa projektissa analysteista oli tullut tiedonvälittäjiä, joiden välillä kehitystiimin jäsenet ja projektiryhmän jäsenet kommunikoivat. Ohjelmiston toimittajan päädyttyä ”Analyst ” keskusteli liiketoiminnan puolella projektin avainhenkilöiden kanssa ja ymmärsi omalla tavallaan mitä halutaan ja hän kirjoitti vaatimusmäärittelystä yksityiskohtaisen dokumentin tai määrittelyn, jonka hän taas kommunikoi kehittäjille ja jonka perusteella kehittäjät alkoivat kehittää ohjelmistoa.

Cohn (2010, 138) onkin todennut, että perinteisessä mallissa analyst voisi todeta ”olen jutellut liiketoiminnan avainhenkilöiden kanssa ja ymmärrän nyt täysin, mitä he haluavat. Kirjoitankin siitä yksityiskohtaisemman dokumentin teitä varten, ja voitte alkaa hommiin”. Scrum-projektissa sama analyst toteaisi ”olen jutellut liiketoiminnan henkilöiden kanssa ja luulen ymmärtäväni ylätasolla, mitä he haluavat. Kirjoitan kuusi käyttäjätarinaa nyt alkuun ja kun meillä tulee lisää kysyttävää liiketoiminnasta, otan teistä muutaman mukaan keskusteluun”.

Scrum-prosessiin liittyvät tapahtumat, roolien uudistukset sekä paremmat työkalut seurata kehitystä ovat parantaneet näkyvyyttä ohjelmiston kehitykseen. Jokainen voi seurata ohjelmiston kehitystä toimivilla projektityökalulla. Työkalujen avulla voi luoda ja seurata käyttäjätarinoita, tarkistaa mitä ominaisuuksia on kehitteillä ja mitä kuluva sprintissä on menossa. Säännölliset palaverikäytännöt on todettu hyvin tarpeellisiksi. Scrum-malliin siirtymisen myötä kommunikaatio on parantunut huomattavasti projektiryhmän ja ohjelmistotuottajan välillä ja tämä on yksi merkittävä tekijä siihen, että myös läpinäkyvyys on parantunut verrattuna aikaisempaan malliin. Ja erityistä positiivista palautetta sai sprinttikatsaustapahtuma, minkä aikana kehittäjät esittelevät projektiryhmälle ja muille sidosryhmille ohjelmistodemon, jolloin on nähtävissä, miten työ edistyy. Sprinttikatsaus edistää myös välitöntä palautteenantoa ja keskustelua kehitetyistä ominaisuuksista. Iso merkitys läpinäkyvyyden kannalta

on, että nähdään varhaisessa vaiheessa toimivaa ohjelmistoa ja sillä, että konkreettinen lopputulos esitetään havainnollistaen mieluummin, kun paperilla esitetyin epä-määräiseen logiikkaan perustuviin luvuin (Auer ym. 2013, 28).

Jokainen projektijäsen ja muutoin ohjelmiston kehityksestä kiinnostunut haluaa tulla katsomaan, mitä ominaisuuksia kehittäjät ovat saaneet aikaan, kuinka teknologia toimii, mitä ominaisuuksia jouduttiin oikomaan, mitä ominaisuuksiin voitiin lisätä ja mitä tulevaan sprinttiin sisältyy. Kun tiimi esittelee sprintin inkrementtejä, auttaa se osallistujia ymmärtämään lisäominaisuuden tuodun hyödyn ja asiakkaalle tuoman arvon. (Schwaber & Beedle 2002, 55–56.) Kun asiakas näkee, että suunniteltuja ominaisuuksia, joita he itse pitävät tärkeimpinä ja hyödyllisimpinä todella toimitetaan, järjestelmä tulee heidän mielissään olemaan todellinen ja he näkevät, mitä arvoa ohjelmisto heille tuottaa. Samaan aikaan kehittäjät tuntevat saavuttaneensa jotain ja että asiakas arvostaa heidän työtään, he ovat entistä motivoituneempia täyttämään asiakkaan tarpeet. (Poppendieck & Poppendieck 2003, 30.)

Tärkeäksi asiaksi koettiin, että tuoteomistajan rooliin nimetty henkilö istuu nimenomaan liiketoiminnan puolella, jossa parhaiten osataan tunnistaa ja määritellä asiakastarpeet ja sen mukaisesti voidaan priorisoida ominaisuuksia ohjelmistoon. Sillä asialla, että tuoteomistaja ei fyysisesti työskentele samassa työympäristössä eikä osallistu päivittäispalaveriinhin kehittäjien kanssa, ei koettu aiheutuneen haasteita, koska tämän päivän modernit viestintäkanavat mahdollistavat tiiviin yhteydenpidon muutoinkin.

### **Scrumin vaikutus laatuun**

Liiketoiminnan määritykset on kannattavampaa kirjoittaa pienemmissä osissa sen sijaan, että laadittaisiin laajat tarkat dokumentit, jossa kaikki asiakkaan tarpeet on jo otettu huomioon. Kirjoitetussa dokumentissa emme tarkista asioiden merkityksiä samalla tavalla kuin tekisimme, jos keskustelisimme asioista. Jos keskustelemme määrittämisestä, voisimme hyvin kysyä, ”että tarkoitatko tällä sitä” tai ”jos ymmärsin oikein,



tämä tarkoittaa” ”eikö tämä vaikuta siihen, että” jne. Kun luemme kirjoitettua dokumenttia, uskomme sen olevan faktaa, emmekä varmista asioita, minkä seurauksena syntyy väärinymmärryksiä. Sen lisäksi kirjoitetut dokumentit vähentävät koko tiimin vastuullisuutta. Niin kuin haastatteluistakin kävi ilmi, on hyvä kuitenkin huomioida, että iso kuva on kuitenkin hahmoteltava ensiksi ja olla ylätasolla tiedossa, jotta voidaan lähteä ominaisuutta pilkkomaan.

Järjestelmältä vaadittujen ominaisuuksien kirjoittaminen käyttäjätarinoiden kautta on paras tapa muuttaa fokus ominaisuuksien tarkasta dokumentoinnista siihen, että ominaisuuksista mieluummin keskusteltaisiin. Kehitystiimi on sitoutunut keskustelemaan tuoteomistajan kanssa ennen kuin aloittavat ominaisuuksien kehittämisen. (Cohn 2009, 238.)

Testaus vesiputousmallissa on jäänyt vähälle huomiolle ja sen seurauksena ohjelmistosta on löytynyt virheitä käyttöön oton jälkeen eikä laatuun olla luonnollisesti oltu tyytyväisiä projektissa. Sen jälkeen, kun kehittäjät ovat toteuttaneet vaaditut toiminnot ohjelmistoon, testaajat ovat päässeet testaamaan ominaisuuksia kuukausien päästä siitä, kun määritykset on toimitettu ohjelmiston tuottajalle. Aikataulut ovat tässä vaiheessa usein olleet tiukilla, jonka seurauksena testausajasta on myös nipistetty.

Scrumin myötä myös testausta on saatu ketterämmäksi. Olennainen ero vesiputousmalliin on, että Scrumissa testaaminen on osa jokaisen erillisen toiminnon toteutusta ja testausta tehdään koko projektin ajan aina sitä mukaa kun toimintoja on saatu ohjelmoitua. Käyttäjätarinoiden hyväksymiskriteerit edellyttävät myös järjestelmän tuottajan testaamaan toiminnot ennen kuin ne tulevat projektiryhmän testattavaksi, joten iso osa virheistä löydetään ja korjataan jo ennen kuin niitä päästetään kehittäjän työpöydältä. Näin ollen koko projektin tasolla aikaa ja vaivaa säästyy samalla kuin kehitettävän ohjelmiston laatu paranee.

Ketterän kehityksen julistuksen (Agile Manifesto 2001) mukaan toimiva ohjelmisto on edistymisen ensisijainen mittari. Tiimi tarkastelee säännöllisesti kuinka voi parantaa tehokkuuttaan ja mukauttaa toimintaansa sen mukaan.

### **Scrumin vaikutus läpimenoaikaan**

Scrum-prosessi perustuu jatkuviin peräkkäisiin kehitysjaksoihin, jolloin ohjelmistoon tuodaan pienemmissä osissa tuotoksia nopealla syklillä. Pienempiä kokonaisuuksien valmistumisaikaa on myös huomattavasti helpompi arvioida ja aikataulut pitävät enemmän paikkaansa. Parannuksia, lisäyksiä ja jopa muutoksiakin voidaan tehdä matkan varrella ja projekti soljuu siitä huolimatta eteenpäin, kun massiivista määrää työtä vesiputousmallin mukaan ole tehty välissä ja aikaa olisi valunut täysin hukkaan, kun pitäisi palata monta vaihetta taaksepäin.

Scrumin tuomia merkittäviä hyötyjä on, että alusta alkaen tuotetaan todellisia tuloksia, jotka testataan ja jotka tämän jälkeen joko hyväksytään tai lähetetään välittömästi korjauksia varten. Aloituspäivän ja käyttökelpoisten tulosten välinen viive on nyt päiviä, ei kuukausia. (Layton 2015, luku 1)

### **Scrumin vaikutus itseohjautuvuuteen**

Itseohjautuvuus on yksi ketterän ohjelmistokehityksen julistuksen periaatteista: parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä (Agilemanifesto 2001).

Itseohjautuvuus on projektijäsenten näkemyksen mukaan näkyvämpää, koska yhteistyötä tehdään tiiviimmin kehittäjien kanssa. Aikaisemmassa vesiputousmallissa, kun kommunikointi oli niin vähäistä kehittäjien kanssa, itseohjautuvuudesta ei saatu oikein käsitystä.

Haastateltavien kokemusten mukaan itseohjautuvuuden lisääntyminen on seurausta myös siitä, että hierarkiat ovat madaltuneet. Scrumin mukaisessa mallissa työskentelevillä ihmisillä on tietyt roolit ja vastuut, mutta ei sellaisia rooleja, jotka pohjautuisivat hierarkkisuuteen. Scrum Masterin, joka toimii erityisesti oman tiiminsä valmentajana ohjatakseen tiimiänsä parempaan yhteistyöhön, tiimihenkeen ja tehokkaampaan suorituskykyyn, tulee olla tehtävän hoitamiseen (Cohn 2010) korostamat tarvit-

tavat kuusi erityispiirrettä: Vastuullisuus, nöyryys, yhteistyökyky, sitoutuminen, vaikuttavuus ja asiantuntemus. Scrum Master huolehtii, että tiimi ymmärtää Scrumin perusteet ja työskentelee optimaalisella tavalla, mutta ei toimi ylätason vahtina, vaan luottaa tiimiinsä ja siihen, että jokainen hoitaa hänelle kuuluvat tehtävät. Scrum-ryhmään palkattu Scrum Master haastateltavien mielipiteiden perusteella todettu onnistuneeksi valinnaksi.

Scrum Masterilla ei ole valtaa sanoa, että ”saat potkut”, vaan hän voi esimerkiksi todeta ”me voimme kokeilla kahden viikon mittaista Sprinttiä ensi kuussa”. Scrum Masteria voisi verrata personal traineriin, joka laatii valmennettavalleen harjoitteluohjelmat ja varmistaa, että kaikki ohjeet toteutetaan ohjeiden mukaan. Hyvä personal trainer myös motivoi ja varmistaa, ettei valmennettava luista annetuista ohjeista ja jätä harjoituksia väliin. Personal trainerin määräysvalta on kuitenkin rajallinen, koska hän ei voi pakottaa valmennettavaa tekemään harjoituksia. Siinä laajuudessa kuin personal trainerilla on vaikutusvaltaa, sen on myöntänyt hänelle asiakas. Myös Scrum Masterilla on vaikutusvaltaa siinä määrin, mitä Scrum-tiimi on hänelle myöntänyt. Scrum Master ei voi todeta, että ”teet niin kuin minä sanon”, kuten projektipäällikkö voisi todeta. (Cohn 2010, 118.)

Mikäli tiimin sisällä on jännitteitä, ne tulisi kitkeä pois, että tiimi voi toimia sujuvasti. Scrumissa opitaan kokemuksista ja tiedostetaan, että kaikki ei välttämättä onnistu heti ja muiden oppimista tuetaan. Scrumissa arvostetaan tiimin oppimiskykyä ja luotetaan tiimin itseohjautuvuuteen. Scrumissa ymmärretään tiimin eri osapuolia ja erilaista asiantuntemusta asioissa. Kaikki ei välttämättä ole niin selvää kaikille ja tiimin jäsenten tuki ja auttamisenhalu ovatkin tärkeitä oppimisessa. Kommunikaatio on avainasemassa ymmärryksen kannalta.

Scrum on parantanut tiimin jäsenten työmotivaalia monella tavalla. Olemalla osa itseohjautuvaa tiimiä antaa ihmiselle mahdollisuuden olla innovatiivinen ja tunnustettu omasta asiantuntemuksestaan. Tiimit voivat järjestää tiimirakenteensa tiettyjen työtapojen ja persoonallisuuksien ympärille, mikä kannustaa ja rohkaisee tiimin jäseniä laajentamaan taitojaan ja osaamistaan, jotta he voivat mahdollistaa liittymisen ha-

luamaansa tiimiin. Itseohjautuvuus kasvattaa myös tiimin suorituskkyä, koska ihmiset, joiden työ on hyvää ja laadukasta haluavat työskennellä yhdessä ja luonnollisesti kääntyvät toistensa puoleen. (Layton 2015.) Tiimin kollektiivisen viisauden hyödyntäminen johtaa usein parempaan tapaan järjestää työt kuin, että se perustuu vain yhden johtajan viisauteen. (Cohn 2010, 189)

Itseohjautuvuus on tänä päivänä yksi johtamisen polttavista puheenaiheista ja kuuluu yhtenä tärkeänä osana myös kohdeyrityksen strategiaan tavoitteisiin. Luukka 2019 korostaa, että itseohjautuvuus tarkoittaa vallan ja päätöksenteon hajauttamista organisaation siten, että johdolla ei ole enää yksityisoikeutta valtaan ja päätöksiin. Käytännössä tämä tarkoittaa sitä, että päätökset tekee aina ihminen tai ryhmä, jolla on olemassa paras tieto päätöksen tekemiseen. Itseohjautuva päätöksenteko edellyttää yrityskulttuurilta myös avoimuutta ja läpinäkyvyyttä, jotta työntekijöillä on pääsy päätöksenteon kannalta oleelliseen tietoon. (Luukka 2019, luku 4). Tarkkaa kontrollia ei tarvita, koska itseohjautuuteen kuuluu, että luotetaan ihmiseen ja hänen haluunsa tehdä asiat hyvin. Lähtökohtana on työntekijöiden vapaus tehdä isoja päätöksiä lupia kyselemättä. (Martela 2015)

Myös Harvard Business Reviewn julkaisussa Agile at Scale 2018 Rigby ja muut kertovat, että johtajat antavat tekemiselle suunnan, mutta eivät kerro sitä, kuinka työt tehdään. Tiiminjäsenet työskentelevät tiiviisti asiakkaan kanssa tai asiakasta edustavien ryhmien kanssa. Tämä asettaa vastuun innovoinnista niiden käsiin, jotka työskentelevät lähimpänä asiakasta ja vastaavasti auttaa vapauttamaan aikaa johtajille luomaan pitkän aikavälin visioita, asettaa strategisia painopisteitä ja rakentaa mahdollisuuksia näiden tavoitteiden saavuttamiseksi (Rigby ym. 2018.)

### **Haasteista johdetut kehitysideat**

Tutkimuksessa korostui tärkeänä huomiona ketteryyden skaalaaminen koko organisaatioon. Ketteriä käytänteitä ja malleja voidaan soveltaa muuhunkin kuin ohjelmistonkehitysprojekteihin. Tämä tarkoittaisi, että organisaatiossa hyödynnettäisiin ketteryyden käytäntöjä, jotta voidaan toimia paremmin muuttuvassa maailmassa. Koh-

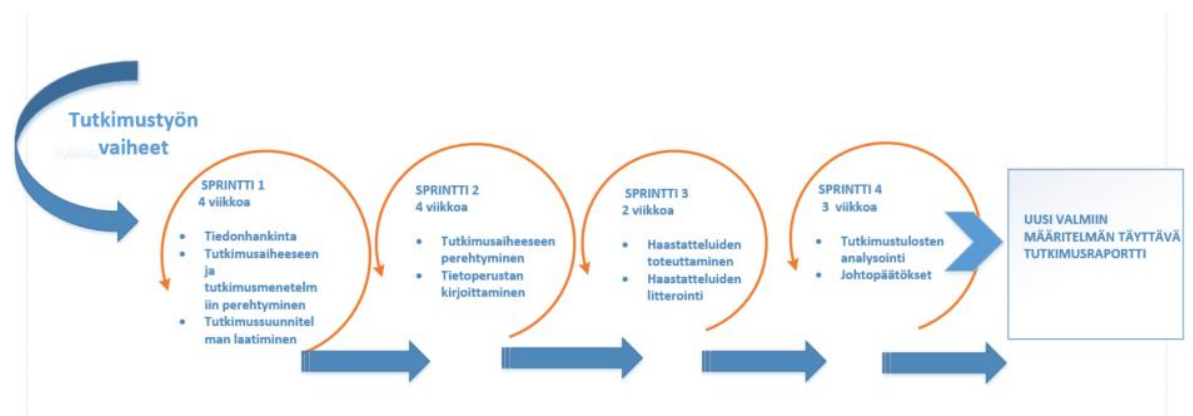
deyrityksessä voitaisiin pohtia, voisiko ketteryyttä laajentaa myös muihin liiketoimintasegmentteihin. Pystyisikö kohdeyritys skaalaamalla ketterää mallia tehostamaan samalla tavalla omaa suorituskykyä kuin Scrum on parantanut yksittäisen ohjelmistonkehitysprojektin suorituskykyä?

Ketterien menetelmien käyttöönotto tuotekehityksessä heijastuu muuhun organisaatioon, jonka toimintatavoissa tulee tarvetta muutokseen. Ketteryyttä tarvitaan myös muulta liiketoiminnalta, josta päätökset voivat tulla hitaasti. Organisaatiossa on hyvä peräänkuuluttaa rohkeaa kokeilun kulttuuria, enemmän kasvokkain keskustelua tarkempien kirjallisten dokumenttien sijaan. Säännöllisesti pidettävät lyhyet palaverit parantavat tiedonkulkua tiimin välillä ja muut ovat tietoisia siitä, mitä muilla on työn alla.

Scrum, projektinhallintamenetelmänä soveltuu projekteihin, jossa kehitetään monimutkaista ja monimuotoista ohjelmistoa, jossa muutokset esimerkiksi asiakkaan muutostarpeiden johdosta ovat välttämättömiä. Mutta Scrumista voi ottaa elementtejä myös moneen muuhunkin projektitoimintaan eli projektin ei tarvitse aina pitää sisällään ohjelmiston kehitystä. Se voi muissa projekteissa tarkoittaa sitä, että projektissa edistettäviä asioita tehdään kahden - neljän viikon kehitysjaksoissa eli sprinteissä. Kun usein tehtävää on paljon ja asiat ovat monimutkaisia, voidaan isommat kokonaisuudet jakaa pienempiin osiin ja asettaa tehtävät lyhyempiin kehitysjaksoihin. Sprintin päätteeksi voidaan tarkastella, mitä on saatu aikaiseksi, onko kaikki tehtävät saatu toteutettua aikataulussa. Mikäli asioita on jäänyt kesken, eikä voida taata, että lopputuotos täyttää valmiin määritelmän, siirtyy tehtävä seuraavaan sprinttiin.

Myös tutkimustyö on oma projektinsa, mikä vaatii vahvaa itsekuria ja itseohjautuvuutta ja projektinhallintataitoja tarvitaan myös tutkimustyöprojektin suunnittelussa, toteutuksessa ja loppuun saattamisessa. Tutkimustyö voidaan myös toteuttaa lyhyissä kehitysjaksoissa pilkkomalla tehtävät osiin, suorittamalla vaadittavat tehtävät aikarajatun sprintin sisällä, tarkastelemalla tehtyä työtä säännöllisesti ja lopputulemana päädytään hyväksytyyn tieteelliseen tutkimusraportin julkaisuun. Sprinttimal-

lisen ajattelun havainnollistamiseksi alla olevassa kuviossa 7 on esitetty, kuinka tämän tutkimuksen työstämisprosessi tapahtui suhteellisen lyhyissä kehitysjaksoissa. Jokainen vaihe edistyi tietyn tuotoksin 2-4 viikon sprinteissä, jotka päättyivät lopulliseen tavoitteeseen eli tutkimusraportin valmistumiseen.



Kuvio 7: Tutkimustyön toteuttaminen sprinteissä

Ketterien tiimien toimintatapoihin kuuluu pitää säännöllisin väliajoin, yksi tai kaksi kertaa kuukaudessa aina sprintin päätteeksi retrospektiivinen palaveri, jossa käydään läpi jakson onnistumiset ja epäonnistumiset ja pohditaan jatkuvasti yhdessä parannuskeinoja. Tällä käytännöllä tavoitellaan tiimin suorituskyvyn parantamista pohtimalla, miten tehdä yksinkertaisemmin ja tehokkaammin asioita.

Kehitystiimien ulkopuolella, muussa kohdeyrityksen tiimitoiminnassa tyypillisesti tiimit kokoontuvat kaksi kertaa vuodessa tiimin kesken ”tiimipäivänä” pohtimaan oman toiminnan parantamista. Myös muut organisaatiossa toimivat tiimit voisivat käydä asioita ja tarkastella omaa toimintaansa dynaamisemmin ja ”retroilla” säännöllisemmin, jolloin tehtävänä olisi tarkastella toimintatapojaan ja muokata tarvittaessa niitä toimivimmiksi, jolloin tiimit oppisivat ja kehittyisivät nopeammin. Jos asioita tarkastellaan vain puolivuositain, usein kerääntyy pitkä lista parannuskohteita ja seuraavan kerran, kun niitä käydään lävitse, useimmat ovat jo voineet unohtaa, mitä aikaisemmassa kehityspäivänä oltiinkaan sovittu kehitettävän. Kehityskohteita on hyvä

tarkastella säännöllisemmin. Ketterässä tiimissä suositetaan tehtävän usein pieniä parannuksia sen sijaan, että keskityttäisiin isoihin parannuksiin harvakseltaan.

Useammat kohdeyrityksessä toimivat tiimit voisivat hyötyä myös kaikille näkyvillä olevasta priorisoidusta työlistasta. Jollakin tiimin jäsenistä olisi Product Owner-tyyppinen vastuu työlistan priorisoimisesta, toki keskustellen ensin tiimin jäsenten kanssa. Jokainen tiimissä työskentelevä olisi tietoinen, mitkä työtehtävät ovat korkeimmalla prioriteetilla työn alla. Jokainen tiimin jäsen tietää omat ja muiden vastuulla olevat työtehtävät ja jokainen pidetään ajan tasalla. Kehitysjakson päätyttyä tiimin kesken pidettäisiin sprinttikatsaus, jossa jokainen esittelee, mitä on saanut työjakson aikana aikaiseksi.

Hyvä toimiva ketterä tiimi tarvitsee hyvän Scrum Masterin, joka toimii erityisesti oman tiiminsä valmentajana ohjatakseen tiimiänsä parempaan yhteistyöhön, tiimihenkeen ja tehokkaampaan suorituskykyyn. Niin kuin Schwaber ja Sutherland (2017) ovat todennet, Scrumin perusteet on helppo omaksua, mutta jokainen ketterä tiimi tarvitsee hyvä Scrum Masterin, joka auttaa kaikkia ymmärtämään Scrumin teorian, käytännöt ja säännöt. Scrum Master vaikuttaa koko Scrum prosessin onnistumiseen ja Scrum Master varmistaa, että palavereihin tullaan hyvin valmistautuneena.

Scrum Masterilla tulee olla jo Scrumin perusteet hallinnassa. Suomessakin järjestetään useita Scrum Master-koulutuksia, jossa saa yksityiskohtaista oppia Scrumista ja Scrumin rakenteesta, jonka avulla voi ohjata ketterää Scrumin käyttöä käytännön työssä. Yrityksessä voitaisiin pohtia, olisiko Scrum Masterin roolille organisaatiossa käyttöä, mikäli Scrum-menetelmää tai sen käytäntöjä halutaan jalkauttaa organisaatiossa laajemmin.

Muuntautuminen uusiin toimintatapoihin ja rooleihin ei kaikille käy aina niin helposti. Mikäli ketteryyttä on tarkoitus laajentaa, muutos on hyvä viedä hallitusti läpi, jotta kaikille syntyy käsitys mitä ollaan muuttamassa ja miksi. Ja niin kuin Rigby ym. (2018) ovat todenneet kaikkia toimintoja ei tarvitse organisoida ketteriksi, koska menetelmät eivät sovi tiettyihin toimintoihin. Mutta jos organisaatiossa perustetaan useita ketteriä tiimejä ja osa jää noudattamaan perinteisempiä menetelmiä, tulee

varmistua siitä, että yhteistyö sujuu ryhmien välillä. Ne ryhmät, jotka eivät toimi ketterästi tulee kuitenkin tukea tiiminä niitä ryhmiä, jotka toimivat ketterien periaatteiden mukaisesti.

## 8 Pohdinta

Kohdeyrityksen ohjelmistokehitysprojektissa ohjelmiston tuottajan toteuttamasta vesiputousmallista saatujen vuosien kokemusten perusteella oli todettu, että vesiputousmainen vaiheittainen prosessimalli ei enää palvelut liiketoiminnan ja asiakkaiden tarpeita. Järjestelmäntuottajan taholla on käyty muutosprosessia läpi, kun vesiputousmallista on siirrytty ketterien menetelmien pariin ja muutoksella on suora vaikutus myös kohdeyrityksen projektiryhmään, jotka tekevät määrityksiä liiketoiminnan, asiakkaiden ja loppukäyttäjien tarpeiden perusteella kehitettävään ohjelmistoon, tekevät loppukäyttäjätestausta kehitetyistä ominaisuuksista ennen tuotosten julkaisua ja käyttöönottoa sekä toimivat tiiviissä yhteistyössä ohjelmistontuottajan kanssa.

Tässä tutkimuksessa tutkittiin kahta projektinhallintamenetelmää perinteistä vesiputousmallia ja ketterän kehityksen projektimallia Scrumia perehtymällä monipuolisesti kirjallisuuteen, aikaisempiin tutkimuksiin, artikkeleihin ja tarinoihin. Tavoitteena tutkimuksen avulla on tuoda esiin mitä hyötyjä ja haasteita projektimallin muutos vesiputouksesta ketterään Scrumiin on tuonut ohjelmistokehitysprojektille projektijäsenten näkökulmasta.

Tutkimuksen tietoperusta on muodostettu aiheeseen liittyvällä teorialla, joka on koostettu tutkimusaihetta koskevasta kirjallisuudesta, luotettavista internet-lähteistä ja tieteellisistä artikkeleista. Tutkimuksellinen osuus toteutettiin laadullisella eli kvalitatiivisella tutkimusmenetelmällä ja tutkimusstrategiaksi valittiin tapaustutkimus. Tapaustutkimus kohdistui kohdeyrityksen yksittäiseen ohjelmistonkehitysprojektiin. Tutkimuksen aineisto kerättiin haastatteleamalla neljää projektiryhmän jäsentä, jotka ovat toimineet pitkään ohjelmistonkehitysprojektissa ja heillä on kokemusta molemmista projektimalleista. Tutkimuksessa korostuukin projektijäsenten oma kokemus



projektihallintamenetelmistä ja uuden mallin tuomista vaikutuksista ja tämän pohjalta koostettiin vertailua projektimallien vesiputouksen ja Scrumin välillä. Teema-haastattelujen analyysistä on nostettu neljä eri teemaa projektimallien vesiputous ja Scrum-menetelmän vertailemiseksi: Läpinäkyvyys, laatu, läpimenoaika ja itseohjautuvuus.

Tarkoituksena tutkimuksessa oli tuoda esiin toimeksiantajalle lyhyen aikavälin näyttöjä ja onnistumisia muutoksesta, joka antaa johdolle kuvaa siitä, onko muutoksella saatu toivottuja hyötyjä. Odotettuja hyötyjä ketterissä menetelmissä ovat tehokkuus, laatu, läpinäkyvyys ja tiimin itseohjautuvuuden parantuminen ja näistä näkökulmista hyötyjä tutkimuksessa tuodaan näkyville. Ketteriä menetelmiä ei ole syytä rajoittaa vain ohjelmistokehitysprojekteihin vaan käytäntöjä voidaan soveltaa myös muussa toiminnassa. Tavoitteena oli myös löytää ja tuoda esiin ketteristä menetelmistä käytänteitä, joita voisivat muutkin ryhmät toteuttaa organisaation sisällä toimiakseen ketterästi.

Haastatteluista kävi ilmi, että kritiikki vesiputousmallia oli hyvin vahvaa ja projektiryhmän jäsenet ovat hyvin tyytyväisiä vaikutuksista, mitä ketterän kehityksen Scrum-menetelmään siirtyminen järjestelmän tuottajan taholla, on heijastanut projektityöhön ja erityisesti yksikön liiketoiminnan kannalta keskeiseen ohjelmistokehitykseen. Scrum-menetelmään siirtyminen koettiin tuoneen parempaa läpinäkyvyyttä ohjelmistokehitysprojektille, säännöllisiä kohtaamisia, mikä on parantanut kommunikation tasoa kehittäjien ja liiketoiminnan välillä sekä lisännyt itseohjautuvuutta verrattuna perinteiseen menetelmään vesiputoukseen. Tuotosten laatu on myös parempaa sekä ominaisuuksia saadaan nopeammin järjestelmään lyhyempisyklisten kehitysjaksojen ansiosta, jolloin tuodaan käyttäjätarinoiden mukaisia pienempiä tuotoksia ohjelmistoon, eivätkä muutoksetkaan hidasta prosessia merkittävästi.

Haastatteluiden perusteella Scrum-menetelmään siirtyminen on tuonut paljon hyviä toimintatapoja, jotka on koettu hyödyllisiksi. Kaikki haastateltavat olivat sitä mieltä, että työskentelevät mieluummin järjestelmäprojektissa Scrum-menetelmällä ja hei-

dän mielestään Scrumin myötä työstä on tullut paljon palkitsevampaa ja motivoivampaa. Muutokset, mitä on tapahtunut Scrum-tyyppiseen toimintamallin siirtymisen myötä muutaman kuukauden aikana, ovat merkittäviä.

Vaikka kommunikaation taso onkin parantunut huomattavasti Scrumiin siirtymisen myötä, nousi haastatteluista esiin haasteena yhteistyö muiden sisäisten ryhmien välillä ja ketteryttä toivottiinkin levitettävän organisaatiossa laajemmin ja hyödynnettävä ketterän kehityksen oppeja. Ketteriä käytäntöjä voidaan soveltaa muuhunkin kuin ohjelmistonkehitysprojekteihin. Toki tämän tyyppistä muutosta ei voi tehdä ennen kuin yritysjohtosta lähtien on ymmärretty ketteryyden periaatteet ja sisäistetty, mitä ketteryys juuri meidän organisaatiossa tarkoittaa.

### **Luotettavuusarviointi**

Tutkimuksen luotettavuuteen kiinnitettiin huomiota jokaisessa tutkimuksen vaiheessa. Tutkimusmenetelmiin perehdyttiin riittävällä tarkkuudella, jotta päästiin selville, mikä menetelmä soveltuu parhaiten tutkittavaan ilmiöön. Tutkimuksen luotettavuutta on pyritty myös kohentamaan perustelemalla jatkuvasti tehtyjä ratkaisuja ottamalla myös kantaa analyysin kattavuuteen ja tehdyn työn luotettavuuteen (Eskola & Suoranta 1998, 209). Hirsjärvi ym. (1997, 232–232) toteavat, että laadullisen tutkimuksen luotettavuutta kohentaa tarkka selostus tutkimuksen kaikkien eri vaiheiden toteuttamisesta. Tämän tutkimuksen aineiston tuottamisen olosuhteet eli paikka, haastatteluiden kesto ja haastatteluiden kulku on selostettu tarkasti ja tutkijan oma arvio aineistonkeruutilanteesta on kuvattu. Tutkimustuloksissa tutkimusselosteita on rikastettu runsaasti suorilla haastatteluotteilla.

Tutkimuksen sisäistä validiutta voidaan arvioida sillä, että tutkimuksessa on tutkittu sitä, mitä on luvattu ja tulkinta on tehty virheettömästi (Kananen 2014, 123). Tämän tutkimuksen tarkoitus oli tutkia kahta projektinhallintamenetelmää sekä mitä hyötyjä ja haasteita projektimenetelmän muutos vesiputousmallista Scrumiin on tuonut ohjelmistonkehitysprojektille projektinjäsenten näkökulmasta. Tutkimus pohjautui projektinhallintamenetelmiä vesiputousta ja Scrumia sekä ketterää kehitystä käsittelevään kirjallisuuteen. Tutkimuksen yhteydessä toteutettiin teemahaastattelut, joiden

teemat ja kysymykset muodostettiin kirjallisuuskatsauksen aihealueiden perusteella ja tutkimustulokset käsiteltiin aineiston analyysissä nousseiden teemojen mukaan. Jotta projektimallin muutoksen tuomia vaikutuksia voitiin tuoda paremmin esille, projektimenetelmistä laadittiin haastatteluihin pohjautuvaa vertailua siitä, kuinka menetelmien luonne, vaiheet ja toimintatavat poikkeavat toisistaan. Näillä perusteilla voidaan katsoa, että tutkimuksessa saavutettiin sisäinen validiteetti.

Ulkoisen validiteetti tarkoittaa tulosten siirrettävyytenä muihin vastaaviin tilanteisiin eli yleistettävyyttä (Kananen 2014, 123). Haastattelurunko on esitetty tutkimuksen liitteenä ja tutkimustuloksista on havaittavissa, että haastateltavien vastaukset olivat hyvin yhteneväisiä, mikä viittaa tutkimuksen toistettavuuteen. Tämän tapaustutkimuksen tutkimustuloksia ei voida kuitenkaan yleistää, koska tutkimuksen kohteena on vain yksi ohjelmistonkehitysprojekti ja haastateltavien määrä oli pieni. Tarkoituksena ei tutkimuksessa ollut luoda yleistyksiä, vaan tuottaa tutkimustuloksia valittua projektia tutkimalla. Näin ollen tutkimuksessa ei saavutettu ulkoista validiteettia.

Eskola ja Suoranta (1998, 18) ovat todenneet luotettavuusarvion kriteeri ei ole kohdejoukon määrä vaan laatu ja (Kananen 2015, 146) puolestaan on todennut, että haastateltaviksi valitaan henkilöt, jotka tietävät parhaiten tutkittavasta ilmiöstä. Aineistoa voidaan pitää kattavana, mikäli vastaukset alkavat toistaa itseään. Tällöin on saavutettu kylläntymispiste eli saturaatio.

Jo toisessa haastattelussa tuli ilmi, että vastauksissa alkoi toistua tiettyjä asioita ja neljännessä haastattelussa ei tullut esiin enää merkittävää uutta tietoa. Haastateltavien mielipiteissä esiintyi hyvin paljon samankaltaisuuksia. Aineistoa voidaan pitää myös sillä perusteella kattavana, että kaikki henkilöt yhtä lukuun ottamatta haastateltiin, ketä voitiin ylipäätään haastatella tutkimusilmiöön liittyen. Haastateltavien määrän lisääminen olisi vääristänyt tutkimustuloksia, koska muut työyhteisön jäsenet eivät työskentele aktiivisesti projektissa eikä heillä olisi ollut riittävää asiantuntemusta ja kokemusta projektinhallintamenetelmistä. Näiden syiden perusteella voidaan todeta, että saturaatio saavutettiin tämän tutkimuksen kohdalla.

Yksi laadullisessa tutkimuksessa käytettävistä luotettavuuskriteereistä on tutkimuksen vahvistettavuus (Kananen 2014, 151). Tutkimustulokset ja tutkimustuloksista muodostettu analyysi toimitettiin haastateltaville luettavaksi ja kommentoitavaksi ennen lopullisen raportin valmistumista. Haastateltaville annettiin mahdollisuus kommentoida tutkimustuloksia ja tuloksista johdettua analyysia ja sen oikeellisuutta. Kaikki haastateltavat vahvistivat tutkimusraportin luettuaan tulkinnan ja kuvauksen oikeaksi, mikä tukee tutkimuksen luotettavuutta tutkitun kannalta.

Tutkimuksen luotettavuuteen saattaa vaikuttaa heikentävästi seikka, että tutkija työskentelee samassa työyhteisössä projektiryhmän kanssa. Tutkimusprosessin jokaisessa vaiheessa pyrittiin kiinnittämään huomiota siihen, ettei tutkija vaikuta tutkimustuloksiin. Aikaisempaa kokemusta ja syvempää tietämystä ei tutkijalla kuitenkaan ollut projektinhallintamenetelmistä ja näin ollen voidaan katsoa, ettei tutkimustuloksiin ollut merkittävää vaikutusta, vaikka tutkija toimii samassa työympäristössä.

Tutkimuksen tietoperustan rakentamiseen ja ilmiön ymmärtämiseen käytettiin monipuolisesti lähteitä, mistä suurin osa koostuu kansainvälisestä kirjallisuudesta. Erityisesti Scrumista kirjallisuutta löytyy kattavasti. Haastavaa oli löytää kirjallisuutta, jossa ketteriä menetelmiä tai Scrum-menetelmää olisi kritisoitu, mihin syynä saattoi vaikuttaa seikka, että useimmat kirjat ovat kirjoittaneet henkilöt, jotka ovat olleet kehittämässä Scrum-menetelmää.

### **Jatkotutkimusaiheet**

Tärkeä jatkotutkimusaihe olisi Scrum-menetelmän vaikutusten seuraaminen pidemmällä aikavälillä ja tarkastelu, ovatko ohjelmistonkehitysprojektissa toimivien henkilöiden tuntemukset vielä samanlaisia esimerkiksi vuoden päästä. Onko Scrumin toimintatavat vakiintuneet, onko jotain muutettu tai lisätty yhteiseen toimintaan. Mielinkiintoinen tutkimusaihe olisi myös tutkia mitä hyötyjä ja haasteita liittyy itse Scrum-tiimissä toimivien kehittäjien ja Scrum Masterin näkökulmasta Scrum-menetelmään. Tutkimuksen yhtenä näkökulmana voisi olla myös asiakastyytyväisyys ja aihepiirin kautta voisi tarkastella, minkälaisin vaikutuksin projektimallin toteutus näkyy asiakkaiden ja ohjelmiston loppukäyttäjien päädyssä. Oleellista olisi selvittää, onko

projektimallin muutoksesta ollut hyötyä asiakkaalle ja onko nopeasyklinen kehitys tuonut arvoa asiakkaalle, jolle ohjelmiston kautta palvelua tuotetaan. Hyötyjen seuranta ja niiden mittaaminen voisi tuoda tutkimukselle määrällisen eli kvantitatiivisen otteen. Mittareina voisivat toimia esimerkiksi asiakkaille lähetettävät tyytyväisyyskyselyt ja asiakaspalveluun saapuvat puhelut ja viestit ohjelmiston toiminnoista ja niiden ratkaisuksista.

## Lähteet

Alasuutari, P. 1994. Laadullinen tutkimus. Tampere: Vastapaino.

Ambler, S., Lines, M. 2012. Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise. Viitattu 2.5.2019. <https://janet.finna.fi/Books24x7ITPro>.

Auer, A., Auer, L., Heinäsmäki, M., Hölttä, J., Kalliala E., Laanti, M., Laine, K., Lekman, L., Miinalainen, P., Naski, H., Piiparinen, T., Puhakka, H., Pyhäjärvi, M., Pääkkönen, T., Räisänen, S., Sora, H., Taipale, M., Talvio, J., Tanninen, A., Toikkanen, T., Toivola, T., Toro, K., Valsta, A., Väyrynen V. & Weissenberg M. 2013. Ketterää kehitystä. 376 vuotta ketteriä kokemuksia. Helsinki: Finn Lectura.

Berkun, S. 2006. Projektihallinnan taito. Suunnittele, hallitse ja johda projekteja. Helsinki: Readme.fi.

Cohn, M. 2010. Succeeding with Agile. Software development using Scrum. Addison-Wesley: Upper Saddle River.

Eskola, J., Suoranta, J. 1998. Johdatus laadulliseen tutkimukseen. Tampere: Vastapaino.

Haikala, I., Märijärvi, J. 2004. Ohjelmistotuotanto. Helsinki: Talentum.

Hirsjärvi, S., Remes, P., Sajavaara, P. 1997. Tutki ja kirjoita. Helsinki: Tammi

Hughes, R. 2016. Agile data warehousing: a guide for solution architects and project leaders. Viitattu 27.2.2019. <https://janet.finna.fi/Record/janet.320272>.

Jones, R. 2007. Project management survival: a practical guide to leading, managing & delivering challenging projects. London: Philadelphia, Kogan Page.

Kananen, J. 2008. Kvali. Kvalitatiivisen tutkimuksen teoria ja käytänteet. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Kananen, J. 2009. Toimintatutkimus yritysten kehittämisessä. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Kananen, J. 2010. Opinnäytetyön kirjoittamisen käytännön opas. Jyväskylä: Jyväskylän ammattikorkeakoulu.

Kananen, J. 2014. Laadullinen tutkimus opinnäytetyönä. Jyväskylä: Jyväskylän ammattikorkeakoulu.

- Kananen, J. 2015. Opinnäytetyön kirjoittajan opas. Näin kirjoitan opinnäytetyön tai pro gradun alusta loppuun. Jyväskylä: Jyväskylän ammattikorkeakoulu.
- Kananen, J. 2017. Kehittämistutkimus interventiotutkimuksen muotona. Opas opinnäytetyön ja pro gradun kirjoittajalle. Jyväskylä: Jyväskylän ammattikorkeakoulu.
- Hammarsten, H. 2016. Ketteryys ei sovi kaikkeen. Artikkelit Aalto University Professional Development-sivustolla. Viitattu 5.5.2019. <https://www.aaltopro.fi/aalto-leaders-insight/2016/ketteryys-ei-sovi-kaikkeen>.
- Layton, M. 2015. Scrum for dummies. Viitattu 26.2.2019. <https://janet.finna.fi/Books24x7>.
- Luukka, P. 2019. Yrityskulttuuri on kuningas. Viitattu 14.4.2019. <https://janet.finna.fi/>.
- Mahalakshmi, M., Sunarajajan, M. 2013. Traditional SDLC Vs Scrum Methodology. Scrum.pdf. Viitattu 25.4.2019. <https://scholar.google.fi/Scrum.pdf>.
- Martela, F. 2017. Itseohjautuvuus - Mistä siitä on pohjimmiltaan kyse.
- Mersino, A. 2013. Emotional intelligence for project managers: the people skills you need to achieve outstanding results. New York: Amacom.
- Myllymäki, R., Hinkka, T., Hirvensalo J., Hämäläinen, J. 2015. Onnistunut tietojärjestelmäprojekti.
- Poppendieck, M., Poppendieck, T. 2003. Lean software Development. Upper Saddle River: Addison-Wesley.
- Pylvänäinen, H. 2009. We are all us: Case study on implementing agile methods in an organization. Opinnäytetyö, ylempi AMK. Jyväskylän ammattikorkeakoulu, yrittäjyyden ja liiketoimintaosaamisen koulutusohjelma. Viitattu 10.03.2019. [https://janet.finna.fi/Record/theseus\\_jamk.10024\\_3589](https://janet.finna.fi/Record/theseus_jamk.10024_3589).
- Rico, D., Hasan, S., Sone, S. 2009. The business value of agile Software methods: Maximizing ROI with Just-In-Time processes and documentation. J. Ross Publishing. Viitattu 14.4.2019. <https://janet.finna.fi/Books24x7>.
- Rigby, D., Sutherland, J., Noble, A. 2018. Harvard Business Review. Agile at Scale. Viitattu 14.4.2019. <https://hbr.org/2018/05/agile-at-scale%20>.
- Rising, L. & Janoff, N. S. 2000. The scrum software development process for small teams. IEEE Software, 17, 4, 26–32. <https://janet.finna.fi/ProQuestABI/INFORM>.
- Schatz, B., Abdalshafi, I. 2005. Primavera Gets Agile: A Successful Transition to Agile Development. IEEE Software, 22, 3, 36–42. <https://janet.finna.fi/ProQuestABI/INFORM>.

Schwaber, K., Beedle, M. 2002. Agile Software Development with Scrum. Upper Saddle River: Prentice Hall cop.

Schwaber, K., Sutherland, J. 2017 Scrum-opas. Scrumin määritelmä ja pelisäännöt. 2017-Scrum-guide-fi-v1-02.pdf. Viitattu 26.2.2019. <https://scrumwell.files.wordpress.com/2018/03/2017-scrum-guide-fi-v1-02.pdf>.

Sininen meteoriitti: Ketteryys haltuun: Yleisimmät ketterät käytännöt. Verkkojulkaisu. Viitattu 22.2.2019. <https://www.meteoriitti.com/2013/06/06/ketteryys-haltuun-yleisimmat-ketterat-kaytannot/>.

Smartsheet: Agile Methology. Viitattu 2.5.2019. <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>.

Suomidigi: Ketterän kehittämisen johtaminen. Viitattu 15.4.2019. <https://suomidigi.fi/menetelmat/ketteran-kehittamisen-johtaminen/>.

The Agile Manifesto. 2001. Viitattu 30.3.2019. [www.agilemanifesto.org](http://www.agilemanifesto.org).

The Standish Group: Chaos Manifesto 2012. Viitattu 5.5.2019. <https://cs.calvin.edu/courses/cs/262/kvlinden/resources/CHAOSManifesto2012.pdf>

Tieturi. 2018. Koulutusmateriaali. Agile\_Requirements\_Development\_v2.0\_uusiTieturi.pdf.

Tuomi, J., Sarajärvi, A. 2009. Laadullinen tutkimus ja sisällön analyysi. Helsinki: Tammi.

Vuori, P. 2014. Ohjelmistoprojektinhallinnan kehittäminen Scrum-menetelmällä. Ylempi AMK opinnäytetyö. Jyväskylän ammattikorkeakoulu, automaatioteknologian koulutusohjelma. Viitattu 27.2.2019. <http://www.theseus.fi/handle/10024/79170>.



## Liitteet

### Ketterään kehityksen takana olevat ydinarvot ja periaatteet (Agile Manifesto 2001)

#### Agile ydinarvot:

- Arvostamme enemmän yksilöitä ja kanssakäymistä kuin menetelmiä ja työkaluja
- Arvostamme enemmän toimivaa ohjelmistoa kuin yksityiskohtaista dokumentointia
- Arvostamme enemmän yhteistyötä asiakkaan kanssa kuin sopimusneuvotteluja
- Arvostamme enemmän vastaamista muutokseen kuin pitäytymistä suunnitelmassa

#### Agile periaatteet:

1. Tärkeintä meille on asiakkaan tarpeiden tyydyttäminen toimittamalla arvoa tuottavia versioita ohjelmistosta aikaisessa vaiheessa ja säännöllisesti.
2. Muutokset myös kehityksen myöhäisemmässä vaiheessa ovat sallittuja. Ketterät menetelmät hyödyntävät muutosta asiakkaan kilpailukyvyyn edistämiseksi.
3. Toimitamme versioita toimivasta ohjelmistosta usein, parin viikon tai kuukauden välein, ja suosimme lyhyempää aikaväliä.
4. Liiketoiminnan edustajien ja ohjelmistokehittäjien tulee työskennellä yhdessä päivittäin koko projektin ajan.
5. Projektit rakennetaan motivoituneiden yksilöiden ympärille. Annamme heille tuen, jonka he tarvitsevat ja luotamme siihen, että he saavat työn tehtyä
6. Tehokkain ja toimivin tapa jakaa tietoa kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu
7. Toimiva ohjelmisto on edistymisen ensisijainen mittari.
8. Ketterät menetelmät kannustavat kestävään toimintatapaan. Kaikkien projektiin osallistujien tulisi pystyä ylläpitämään tasainen työtahti.
9. Teknisen laadun ja ohjelmiston hyvän rakenteen jatkuva huomiointi auttaa projektia pysymään ketteränä.
10. Yksinkertaisuus - tekemättä jätettävän työn maksimointi - on tärkeää.
11. Parhaat arkkitehtuurit, vaatimukset ja suunnitelmat syntyvät itseorganisoituvissa tiimeissä.
12. Tiimi tarkastelee säännöllisesti, kuinka parantaa tehokkuuttaan, ja mukauttaa toimintaansa sen mukaisesti.

## Teemahaastattelurunko:

### Johdatus ja haastateltavan taustatiedot

- Tutkimuksen tarkoitus ja luottamuksellisuus
- Työkokemus
- Kuinka monta vuotta työskennellyt projektissa
- Rooli ja työtehtävät projektissa

### Projektijäsenten näkemys työskentelystä vesiputousmenetelmällä

- Minkälaisia rooleja oli mukana prosessissa?
- Minkälaisia hierarkioita rakentui prosessin ympärille?
- Kuinka liiketoiminnan vaatimusmäärittelyä tehtiin?
- Minkälaista dokumentaatiota syntyi?
- Kuinka kommunikointi sujui järjestelmän tuottajan ja liiketoiminnan välillä?
- Minkälaiset palaverikäytännöt olivat käytössä?
- Millä tavalla toteutuksen aikatauluja arvioitiin?
- Miten järjestelmän tuottaja suhtautui muutoksiin?
- Miten projektin etenemistä seurattiin?
- Millaista oli laatu ja ohjelmiston käytettävyys julkaisujen jälkeen?
- Mitä muita hyötyjä ja haasteita, mitä haastattelun aikana ei tullut vielä esiin?

### Projektijäsenten näkemys Scrumin käyttöönotosta ja muutoksen aikaansaamista vaikutuksista

- Minkälaisia rooleja on mukana prosessissa?
- Minkälainen vaikutus uusilla rooleilla on projektityöskentelyyn?
- Miten kommunikaatio sujuu liiketoiminnan ja järjestelmän tuottajan välillä?
- Mitä hyötyjä ja haasteita on siinä, että vaatimusmäärittelyt kirjoitetaan pienempinä palasina isojen kokonaisuuksien sijaan?
- Minkälaista dokumentaatiota syntyy?
- Minkälainen vaikutus säännöllisillä palaverikäytännöillä on projektityöskentelyyn?
- Millä tavalla aikatauluja arvioidaan? Ovatko ne paikkansapitäviä?
- Miten järjestelmän tuottaja suhtautuu muutoksiin?
- Miten projektiryhmän jäsenet ovat tietoisia projektin etenemisestä?
- Millaista on laatu ja järjestelmän käytettävyys julkaisujen jälkeen?
- Mitä muita hyötyjä projektimallin muutoksesta on seurannut, mitä haastattelun aikana ei ole tullut vielä esiin?
- Mitä haasteita on tullut vastaan Scrumiin siirtymisen myötä