

Reeta Häkkinen

**PROJEKTIPÄÄLLIKÖN KOMPETENSSIT  
HAJAUTETUSSA KETTERÄSSÄ  
OHJELMISTOKEHITYKSESSÄ**



JYVÄSKYLÄN YLIOPISTO  
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA  
2020

## TIIVISTELMÄ

Häkkinen, Reeta

Projektipäällikön kompetenssit hajautetussa ketterässä ohjelmistokehityksessä

Jyväskylä: Jyväskylän yliopisto, 2020, 45 s.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja: Marttiin, Pentti

Monissa yrityksissä on 2000-luvun aikana siirrytty käyttämään ketteriä menetelmiä ohjelmistoprojektien toteutuksessa. Samalla merkittävä osa projekteista toteutetaan hajautetusti eri toimipisteissä tai alihankintana eri puolilla maailmaa. Näiden kahden toimintatavan yhdistäminen tuo ohjelmistoprojekteihin erinäisiä haasteita, sillä ketterät menetelmät painottavat osittain sellaisia asioita, joita on hajautetussa ympäristössä haastavaa toteuttaa. Ketterissä menetelmissä korostetaan esimerkiksi aktiivisen kasvokkain tapahtuvan kommunikoinnin merkitystä, mikä on hajautetuissa projekteissa merkittävä haaste aikaerojen ja maantieteellisen etäisyyden vuoksi. Ketterässä lähestymistavassa arvostetaan myös tiimien itseohjautuvuutta ja dokumentaatio pyritään pitämään mahdollisimman vähäisenä, kun taas hajautetussa ympäristössä on kommunikoinnin rajoitusten seurauksena enemmän tarvetta dokumentoinnille sekä tiimien koordinoinnille. Tässä tutkimuksessa selvitetään, millaisia haasteita ketterien menetelmien käyttö tuo hajautettuihin ohjelmistoprojekteihin, sekä millaista osaamista projektipäälliköltä vaaditaan erityisesti hajautettujen ketterien projektien johtamisessa. Tutkimus on toteutettu kirjallisuuskatsauksena. Tulosten perusteella hajautettujen ketterien ohjelmistoprojektien haasteet liittyvät pääosin kommunikaatioon, koordinointiin ja luottamukseen sekä etäisyyden mukanaan tuomiin haasteisiin, kuten kulttuuriin ja aikaeroihin. Näiden haasteiden pohjalta projektipäällikön taidoista hajautetuissa ketterissä ohjelmistoprojekteissa korostuvat erityisesti kommunikointiin, tiimijohtamiseen, projektinhallintaan sekä organisointiin liittyvät taidot. Tutkielman tuloksia voidaan hyödyntää projektipäällikön osaamisen kehittämisessä sekä pohdittaessa projektin toteutustapoja, kuten kannattaako projektia toteuttaa hajautetusti ja millaisia käytänteitä tällöin kannattaa suosia.

Asiasanat: hajautettu ketterä ohjelmistokehitys, ketterä kehitys, Scrum, projektipäällikön kompetenssit, projektin hallinta

## **ABSTRACT**

Häkkinen, Reeta

Project manager competencies in distributed agile development

Jyväskylä: University of Jyväskylä, 2020, 45 pp.

Information Systems, Bachelor's Thesis

Supervisor: Marttiin, Pentti

In the 21st century many corporations have started using agile methods for implementing software development projects. At the same time a significant amount of software development projects are implemented in distributed environments or outsourced globally. Combining these two methods creates a multitude of challenges for software development projects due to agile methods emphasizing practices that are difficult to implement in a distributed environment. For example, agile methods emphasize the importance of face-to-face communication which in turn is a significant challenge in distributed development resulting from time differences and geographical distance. Agile approach also values self-organizing teams and aims to keep documentation to a minimum whereas in distributed development there is more need for documentation and coordinating the teams due to the restricted ability to communicate. This study aims to determine the challenges the use of agile methods brings into distributed software development projects along with what kind of competences are required from a project manager especially in leading distributed software development projects. The study is implemented as a literature review. Based on existing literature the challenges in distributed agile development projects are mainly related to communication, coordination and trust along with challenges related to distance, such as culture and time differences. Derived from these challenges the skills required from a project manager are related to communication, team management, project management and organization. The results of this study can be utilized in the competence development of project managers and while trying to determine if implementing the project as distributed is worthwhile and what kind of practices should be favoured.

Keywords: distributed agile development, agile development, Scrum, project manager competence, project management

## KUVIOT

KUVIO 1 Hajautetun Scrumin mallit Sutherlandin ym. mukaisesti.....	19
KUVIO 2 Hajautetun ketterän ohjelmistokehityksen haasteet .....	21

## TAULUKOT

TAULUKKO 1 Hajautetun ketterän ohjelmistokehityksen haasteet Kajko-Mattssonin ym. mukaan .....	23
TAULUKKO 2 IT-alan projektipäällikön taidot kategorioittain.....	29
TAULUKKO 3 Projektipäällikön keskeisimmät taidot.....	31
TAULUKKO 4 Rameshin ym. hajautettujen ketterien projektien haasteet sekä niihin vaikuttavat projektipäällikön taidot.....	34

# SISÄLLYS

TIIVISTELMÄ .....	2
ABSTRACT .....	3
KUVIOT .....	4
TAULUKOT .....	4
1 JOHDANTO.....	7
2 KETTERÄ OHJELMISTOKEHITYS .....	10
2.1 Ketterän ohjelmistokehityksen periaatteet .....	10
2.2 Scrum.....	12
2.2.1 Scrum-tiimin roolit.....	12
2.2.2 Scrumin tapahtumat .....	13
2.2.3 Scrumin artefaktit.....	15
3 HAJAUTETTU KETTERÄ OHJELMISTOKEHITYS.....	16
3.1 Hajautettu ohjelmistokehitys .....	17
3.2 Ketteryyden yhdistäminen hajautettuun ohjelmistokehitykseen .....	18
3.3 Hajautettu Scrum .....	18
3.4 Hajautetun ketterän ohjelmistokehityksen haasteet.....	20
3.4.1 Hajautetun ketterän ohjelmistokehityksen haasteet Rameshin ym. (2006) mukaan.....	20
3.4.2 Hajautetun ketterän ohjelmistokehityksen haasteet muissa tutkimuksissa .....	22
3.4.3 Etäisyys .....	25
4 PROJEKTIPÄÄLLIKÖN KOMPETENSSIT .....	26
4.1 Projektipäälliköiden koulutus ja sertifikaatit .....	26
4.2 IT-alan projektipäällikön kompetenssit.....	27
4.3 IT-alan projektipäällikön taidot Keilin ym. (2013) mukaan .....	28
5 PROJEKTIPÄÄLLIKÖN TAIDOT OSANA HAJAUTETTUJEN KETTERIEN MENETELMIEN HAASTEISIIN VASTAAMISTA.....	32
5.1 Projektipäällikön rooli ketterissä projekteissa.....	32
5.2 Projektipäällikön taitojen merkitys hajautettujen projektien haasteiden ratkaisussa .....	33
6 YHTEENVETO .....	38

LÄHTEET .....	42
---------------	----

# 1 JOHDANTO

Ohjelmistokehityksessä on 2000-luvulla siirrytty yhä vahvemmin ketterien menetelmien käyttöön, ja samalla globalisaation seurauksena merkittävä osa ohjelmistokehitysprojekteista toteutetaan hajautetusti eri toimipisteiden ja maiden välillä, joko ulkoistamalla tai siirtämällä konsernin sisäisesti toimintoja toiseen maahan. Suurimmat syyt hajautetun ohjelmistokehityksen suosiolle ovat kustannussäästöt sekä mahdollisuus laajempaan tarjontaan työntekijöitä ja osaamista (Carmel & Agarwal, 2001). Matalonga, Solari ja Matturro (2013) nostavat esiin myös tarpeen päästä hajautuksella lähemmäksi asiakasta ja mahdollisuuden mukauttaa tuotetta paremmin paikalliseen kulttuuriin. Lisäksi hajautus antaa mahdollisuuden ympärivuorokautiseen kehitykseen ja tätä kautta myös nopeampaan julkaisuun (Matalonga, Solari & Matturro, 2013).

Ketterien menetelmien suosio sen sijaan perustuu ketterän ohjelmistokehityksen arvoihin, jotka korostavat kommunikaatiota, muutoksiin vastaamista, toimivaa ohjelmistoa sekä asiakasyhteistyötä. Ketteristä menetelmistä esimerkiksi Scrum on yksi suosituimmista projektinhallinnan viitekehyksistä. Ketterissä malleissa kehitystä tehdään lyhyissä iteraatioissa ja asiakkaalle toimitetaan säännöllisesti uusia, toimivia versioita ohjelmistosta.

Yhä useammin nämä kaksi lähtökohtaa, ketterät menetelmät sekä projektin hajautus, yhdistetään samaan hankkeeseen. Ketterät menetelmät ja hajautettu ohjelmistokehitys eivät kuitenkaan periaatteiltaan ole yhteneväisiä, sillä esimerkiksi ketterien menetelmien korostama kasvokkain tapahtuva kommunikointi on hajautetussa ympäristössä haastava toteuttaa, ja se vaatii matkustamista toimipisteiden välillä. Myös dokumentaation tarve sekä koordinoinnin määrä poikkeavat tyypillisesti toisistaan ketterissä malleissa ja hajautetussa ohjelmistokehityksessä. Hajautetussa ympäristössä on yleensä tarvetta kattavalle koordinoinnille niin tiimien kuin niiden jäsenten välillä, ja suurempi määrä dokumentaatiota puolestaan korvaa kommunikaation aiheuttamia rajoituksia. Ketterissä menetelmissä sen sijaan suositaan mahdollisimman vähäistä dokumentaatiota sekä tiimien itseohjautuvuutta.

Nämä ristiriidat aiheuttavat haasteita ketterien projektien toteuttamiseen hajautetussa ympäristössä, ja tämän seurauksena myös projektipäällikön tulee

olla tietoinen haasteista, joita tällaisissa projekteissa tyypillisesti kohdataan. Tärkeää on myös tunnistaa keinoja selvittää ongelmatilanteista. Projektipäällikön taidoilla onkin kriittinen merkitys IT-alan projektien onnistumisen kannalta (Keil, Lee & Deng, 2013). Tämän vuoksi on erittäin tärkeää tiedostaa, mitkä projektipäällikön taidot ovat keskeisimpiä projektin kannalta ja pyrkiä kehittämään erityisesti näitä taitoja. Hajautettu ketterä ohjelmistokehitys eroaa käytänteiltään ja haasteiltaan perinteisistä projekteista, mikä asettaa myös projektipäällikön taidoille erilaista painoarvoa perinteisiin IT-alan projekteihin verrattuna.

Tässä tutkielmassa pyritään selvittämään, mitkä ovat projektipäällikölle tarpeellisia kompetensseja erityisesti tilanteessa, jossa ketterä ohjelmistokehitysohjelma toteutetaan hajautettuna. Varsinaiset tutkimuskysymykset ovat:

- Mitä haasteita hajautetussa ketterässä ohjelmistokehityksessä on ja mitä niiden korjaamiseksi voidaan tehdä?
- Mitkä taidot ja osaamisalueet ovat IT-alan projektipäällikölle erityisen tärkeitä hajautetussa ketterässä ohjelmistoprojektissa?

Tutkimus on toteutettu kirjallisuuskatsauksena. Kirjallisuutta on haettu seuraavilla hakusanoilla sekä niiden yhdistelmillä: "distributed agile development", "distributed software development", "global software development", "Scrum", "project manager skills" ja "project manager competences". Aineistoa on haettu pääasiassa Google Scholar -hakukoneen avulla sekä IEEE-tietokannasta. Tutkielmassa käytettävää aineistoa on rajattu siten, että projektipäällikön osaamista ja kompetensseja koskeva aineisto on rajattu ainoastaan IT-alan projektipäälliköitä koskevaan tutkimukseen, ja hajautettuja projekteja koskeva aineisto on rajattu kattamaan ainoastaan ohjelmistoprojekteja sekä muita IT-alan projekteja. Rajaus on tehty artikkelien tiivistelmien perusteella sekä silmäilemällä artikkelien sisältöä. Lähteiden rajauksessa on otettu huomioon myös lähteiden laatu sekä julkaisuajankohta. Lähdeaineistossa on pyritty hyödyntämään vertaisarvioituja tieteellisiä artikkeleita alan lehdistä sekä konferenssijulkaisuja.

Tutkimustulosten perusteella hajautetun ketterän ohjelmistokehityksen haasteet koskevat kommunikointia sekä kontrollin ja luottamuksen puutetta. Nämä aiheuttavat ristiriitoja kommunikoinnissa, laatuvaatimuksissa, ohjauksessa ja sopimuksissa sekä vähentävät tiimin yhteenkuuluvuuden tunnetta (Ramesh, Cao, Mohan & Xu, 2006). Näiden haasteiden korjaamiseksi hajautetuissa ketterissä projekteissa tulisi erityisesti kiinnittää huomiota viestinnän kehittämiseen, tietämyksen jakamisen edistämiseen, prosessin jatkuvaan mukauttamiseen sekä luottamuksen rakentamiseen (Ramesh ym., 2006). Projektipäällikön osaamisalueista näiden osalta korostuu erityisesti kommunikointi, tiimijohtaminen, organisointi ja projektin hallinta, sekä liiketoiminta-alueen tuntemus ja laatu keskeisyys.

Tätä projektipäällikön tärkeimpien taitojen listaa voidaan hyödyntää monin tavoin. Yritykset voivat hyödyntää listausta uusien projektipäälliköiden



rekrytoinnissa ja peilata hakijoiden osaamista siihen, mitkä taidot ovat hajautetuissa projekteissa erityisen merkittäviä. Toisaalta listaus auttaa myös organisaatioita ja kouluttajia suuntamaan koulutustarjontaa siten, että se antaa valmiuksia hajautetuissa ketterissä projekteissa toimimista varten. Tämän tutkimuksen tuloksista on hyötyä myös IT-alan projektipäälliköille itselleen, sillä se auttaa hahmottamaan oman osaamisen tilaa sekä mahdollista tarvetta koulutautua ja kehittyä lisää hajautettujen projektien johtamisen saralla.

Tutkimuksen tuloksia voidaan hyödyntää myös, kun pohditaan, onko projektia tarpeen toteuttaa hajautettuna, vai ovatko siitä johtuvat haasteet suurempia kuin hajautuksen tuomat hyödyt. Tutkimus tarjoaa myös ratkaisukeinoja siihen, mitä ketterästä kehityksestä kannattaa erityisesti hyödyntää hajautettujen projektien yhteydessä, jotta päästäisiin haluttuun lopputulokseen.

Tutkielma koostuu neljästä sisältöluvusta sekä yhteenvedosta. Tutkielman toisessa luvussa käsitellään ketterää ohjelmistokehitystä sekä sen tärkeimpiä periaatteita, ja esimerkkinä ketteristä menetelmistä esitellään Scrum. Kolmannessa luvussa käydään tarkemmin läpi ketterän ohjelmistokehityksen soveltamista hajautettuun ympäristöön, esitellään hajautetun Scrumin malli ja eritellään hajautuksen tuomia haasteita sekä mahdollisia ratkaisukeinoja. Neljännessä luvussa keskitytään IT-alan projektipäällikön kompetensseihin ja esitellään Keilin ym. (2013) malli projektipäällikön tärkeimmistä taidoista ja ominaisuuksista. Viidennessä luvussa verrataan projektipäällikön kompetensseja ja hajautetun ketterän ohjelmistokehityksen haasteita toisiinsa ja pyritään selvittämään, mitkä projektipäällikön taidot voisivat erityisesti olla yhteydessä ketterän projektin hajautuksen tuomien haasteiden ratkaisuun. Kuudes luku on yhteenvedo, jossa kootaan tutkimuksen tulokset sekä esitellään jatkotutkimusaiheet.

## 2 KETTERÄ OHJELMISTOKEHITYS

Luvussa käydään läpi yleisesti ketterien menetelmien periaatteita sekä esitellään tarkemmin jo pitkään suosiossa ollut ketterä menetelmä, Scrum. Ketterien menetelmien periaatteet ja ketterän ohjelmistokehityksen julistus käydään läpi alaluvussa 2.1 ja Scrumia sekä sen käytänteitä käsitellään alaluvussa 2.2. Scrumin esittely perustuu viimeisimpään, vuonna 2020 julkaistuun versioon Schwaberin ja Sutherlandin (2020) Scrum Guidesta.

### 2.1 Ketterän ohjelmistokehityksen periaatteet

Ketterät menetelmät ovat nousseet 2000-luvun alun aikana huomattavaan suosioon ohjelmistokehitysprojekteissa. Ketterään ohjelmistokehitykseen viittaavia toimintatapoja alkoi kuitenkin kehittyä jo 1990-luvulla, kun nähtiin tarve vaihtoehtoisille ratkaisuille perinteisten raskaiden ohjelmistokehityksen toimintamallien sijaan. Perinteiset ohjelmistokehityksen mallit aiheuttivat haasteita erityisesti tilanteissa, joissa projektin vaatimukset muuttuivat kesken projektin läpiviennin ja ohjelmistokehitykseltä alettiin kaivata enemmän dynaamisuutta (Lee & Xia, 2010). Tämän seurauksena 1990-luvun lopulla alettiin suunnitella ensimmäisiä ketteryyttä korostavia malleja, ja muun muassa Schwaber esitteli vuonna 1997 Scrum-mallin (Schwaber, 1997).

Vuonna 2001 julkaistiin neljään ydinarvoon ja 12 periaatteeseen pohjautuva Ketterän ohjelmistokehityksen julistus (Agile manifest). Julistuksen neljä ydinarvoa ovat:

Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja.

Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota.

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja.

Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa.

(Agile Alliance, 2001).

Ketterän ohjelmistokehityksen julistuksen periaatteet täydentävät neljää ydin-arvoa. Ketterän kehityksen periaatteiden mukaisesti asiakkaalle pyritään toimitamaan versioita toteutettavasta ohjelmistosta jo aikaisessa vaiheessa projektia, sekä säännöllisesti koko projektin ajan. Ohjelmistoversioiden toimittamisessa suositaan lyhyttä aikaväliä, parista viikosta pariin kuukauteen. (Agile Alliance, 2001).

Kommunikoinnin suhteen ketterässä ohjelmistokehityksessä suositaan kasvokkain tapahtuvaa kommunikointia, sillä se koetaan tehokkaimmaksi tavaksi välittää tietoa projektin jäsenten välillä. Myös liiketoiminnan edustajien ja ohjelmistokehittäjien välisen jatkuvan, jopa päivittäisen yhteydenpidon merkitystä korostetaan. (Agile Alliance, 2001).

Ketterät menetelmät mahdollistavat vaatimusten muuttamisen kesken projektin, mikä nähdään tärkeänä osana asiakkaan kilpailukyvyn edistämistä. Ylipäätään tärkeimpänä edistystä mittaavana tekijänä pidetään toimivaa ohjelmistoa, ja tuotteen tekniseen laatuun sekä ohjelmakoodin rakenteeseen pyritään kiinnittämään huomiota, sillä sen koetaan edistävän ketteryyttä. Periaatteissa korostetaan myös toteutuksen yksinkertaisuutta, jotta turhan työn tekeminen vähenee. (Agile Alliance, 2001).

Tiimin ja sen yksilöiden toiminnan merkitys on tärkeässä osassa ketterää ohjelmistokehitystä. Tiimin jäsenten motivaatio nostetaan tärkeäksi periaatteeksi, jota vastaan luottamus työn toteutumiseen on vahvempi. Tiimeiltä odotetaan itseorganisointumista, mikä voi nostaa arkkitehtuurivalintojen, vaatimusten sekä suunnitelmien laatua. Tiimin toiminnasta nostetaan esiin myös tehokkuuden parantaminen ja oman toiminnan mukauttaminen. Projektitiimille tulee tarjota tukea ja tarvittavat puitteet työskentelylle. On tärkeää myös ylläpitää kestävää toimintatapaa, jotta hankkeen osapuolet pystyvät säilyttämään työtahtinsa myös tulevaisuudessa. (Agile Alliance, 2001).

Vaikka ketterät menetelmät ovat saavuttaneet suurta suosiota, on niiden soveltaminen laajoissa projekteissa ollut jo pidemmän aikaa haaste. Saeedan, Arifin, Minhasin ja Humayunin (2015) mukaan ketterät menetelmät sopivat erityisesti pieniin ja keskisuuriin projekteihin, mutta laajoissa projekteissa ketterät menetelmät voivat aiheuttaa monia haasteita. Ketterien menetelmien skaalautumiseen vaikuttavia tekijöitä Saeedan ym. (2015) mukaan ovat tiimin koko, maantieteellinen hajautuminen, sääntöjen noudattaminen, organisaation kompleksisuus, tekninen kompleksisuus, organisatorinen hajautuminen, toimialan kompleksisuus sekä yrityksen periaatteet. Ketterien menetelmien skaalautumista voidaan edistää jakamalla laajoja projekteja pienemmiksi osaprojekteiksi (Saeeda ym., 2015). Usein laajoissa projekteissa onkin yhden ison tiimin sijaan useita pienempiä tiimejä. Tämä antaa mahdollisuuden kontrolloida projektia tehokkaammin, mutta toisaalta se lisää haasteita johtamisen tasolla, kun projektien koordinointi vaatii enemmän huomiota. (Petersen & Wohlin, 2009).

## 2.2 Scrum

Scrum on viitekehys, jonka tarkoituksena on toimia tiimien ja projektien apuvälineenä monimutkaisten tuotteiden kehittämisessä. Kantavana ajatuksena Scrumissa on näkemys siitä, että tiedon pohjana toimii kokemus ja päätöksenteko perustuu tehtyihin havaintoihin. Ennustettavuutta ja riskien hallintaa tuetaan iteratiivis-inkrementaalisella lähestymistavalla, mikä käytännössä tarkoittaa toiminnallisuuden toteuttamista pienissä osissa lyhyiden, toistettavien työskentelyjaksojen (sprinttien) aikana. (Schwaber & Sutherland, 2020).

Scrumin pohjana on kolme merkittävää tekijää: läpinäkyvyys, tarkastelu ja sopeuttaminen. Läpinäkyvyydellä viitataan projektin lopputuloksesta vastaavien tahojen tarpeeseen havaita prosessin kannalta merkittävät tekijät. Mikäli näitä tekijöitä ei ole määritelty selkeästi, läpinäkyvyys kärsii ja tarkastelijoilta puuttuu yhtenäinen näkemys tarkasteltavasta asiasta. Tarkastelulla tarkoitetaan säännöllistä Scrumin tuotosten sekä edistymisen seuraamista, mikä on tarpeen, jotta mahdolliset ongelmat havaitaan aikaisessa vaiheessa. Tarkastelua varten Scrumissa on tapahtumia (sprintti, sprintin suunnittelu, päivittäispalaveri, sprintin katselmointi, retrospektiivi), joissa tuotoksia ja edistymistä voidaan seurata. Mikäli jokin prosessin osista ei täytä sille asetettuja rajoja ja sen seurauksena tuotetta ei voi hyväksyä, on tällaisessa tilanteessa tärkeää mukauttaa prosessia mahdollisimman nopeasti, jotta myöhempien poikkeamien määrä saada pidettyä mahdollisimman vähäisenä. (Schwaber & Sutherland, 2020).

### 2.2.1 Scrum-tiimin roolit

Scrum-tiimi muodostuu scrummasterista, tuoteomistajasta sekä kehittäjistä. Tiimissä ei ole hierarkiaa, eikä se jakaudu pienempiin alitiimeihin. Kaikki tarvittava osaaminen löytyy tiimin sisältä ja se kykenee toimimaan itseohjautuvasti. Esimerkiksi tehtävien jaon tiimi tekee ilman ulkopuolista ohjausta. (Schwaber & Sutherland, 2020).

Tiimin koko pyritään pitämään tarpeeksi pienenä, jotta ketteryys säilyy, mutta toisaalta myös riittävän suurena, että vaadittavat tehtävät saadaan sprintin aikana toteutettua. Tyypillisesti tiimin koko on alle 10 henkilöä, mutta mikäli on tarvetta suuremmalle tiimille, on vaihtoehtona luoda useita rinnakkaisia tiimejä, joilla on sama tavoite sekä yhteinen työjono ja tuoteomistaja. Tiimien pienen koon etuna on tehokkuus ja aktiivisempi kommunikointi tiimin jäsenten välillä. (Schwaber & Sutherland, 2020).

Tiimin vastuulla on kaikki tuotteeseen liittyvä toiminta, kuten yhteistyö sidosryhmien kanssa, ylläpito, toimitus ja tuotekehitys. Koko tiimi vastaa myös siitä, että sprintin aikana saadaan toteutettua julkaisukelpoinen inkrementti eli tuoteversio. (Schwaber & Sutherland, 2020).

Kehittäjien vastuulla on suunnitella jokainen sprintti, minkä lisäksi he vastaavat toteuttamiskelpoisen inkrementin luonnista sprintin aikana. Tuoteomistaja sen sijaan vastaa kehittäjien toteuttaman tuotteen arvon maksimoinnista ja

tuotteen kehitysjonon hallinnasta. Tuotteen kehitysjonon (product backlog) hallintaan kuuluu muun muassa varmistaa, että kehitysjonon kohdat ovat selkeästi ilmaistuja ja ymmärrettäviä sekä järkevissä järjestyksessä tavoitteiden kannalta. Tuoteomistaja voi myös delegoida tehtävät kehitystiimille, mutta vastuu pysyy siitä huolimatta tuoteomistajalla. (Schwaber & Sutherland, 2020).

Scrummasterin tehtävänä on varmistaa, että projektia edistetään Scrumin käytäntöjen ja sääntöjen mukaisesti, ja että tiimi toimii tehokkaasti. Scrummasterin rooli on olla palveleva johtaja, joka auttaa tiimiä parantamaan toimintatapojaan Scrumin viitekehyksen mukaisesti, ohjaa tiimin jäseniä itseohjautuvuuteen, poistaa esteitä kehitystiimin edistymisen tieltä sekä varmistaa, että Scrumin tapahtumat pidetään sovittuina aikoina. Scrummaster palvelee myös tuoteomistajaa muun muassa auttamalla tavoitteen määrittelyssä sekä kehitysjonon hallinnassa. Tuoteomistajan sekä kehittäjien auttamisen lisäksi scrummaster palvelee osaltaan myös organisaatiota: hän vastaa Scrumin käyttöönotosta sekä suunnittelee ja neuvoa Scrumin toteuttamisessa. (Schwaber & Sutherland, 2020). On tärkeää huomata, että scrummaster ei kuitenkaan tarkoita projektipäällikköä, vaikka joissakin tapauksissa projektipäällikkö voi toimia myös scrummasterina.

### 2.2.2 Scrumin tapahtumat

Scrumissa on erilaisia tapahtumia, joiden tarkoituksena on lisätä tuotekehityksen kannalta tärkeää läpinäkyvyyttä sekä antaa enemmän mahdollisuuksia tarkasteluun. Näitä tapahtumia ovat sprintin suunnittelu, päivittäispalaverit, sprintin katselmus sekä retrospektiivi. Tapahtumat lisäävät säännöllisyyttä ja vähentävät tarvetta muille palaverille. (Schwaber & Sutherland, 2020).

Scrum koostuu lyhyistä, maksimissaan kuukauden pituisista sprinteistä, joiden aikana suunnitelmien pohjalta luodaan arvoa tuottavia inkrementtejä eli tuoteversioita. Edellisen sprintin loputtua uusi sprintti aloitetaan välittömästi, ja yhden sprintin aikana pidetään aina kyseisen sprintin suunnittelupalaveri, päivittäiset palaverit, sprintin katselmointi sekä retrospektiivi. Sprintin aikana ei tehdä lainkaan sellaisia muutoksia, jotka voisivat vaarantaa kyseiselle sprintille asetetun tavoitteen saavuttamisen, eikä jo asetetuista laatutavoitteista muutoinakaan tingitä. Sen sijaan kehitysjonoa voidaan sprintin aikana tarpeen mukaan muokata ja sprintin sisältöä tarkentaa, kunhan asiasta keskustellaan tuoteomistajan kanssa. (Schwaber & Sutherland, 2020).

Sprinttien hyötynä on niiden tuoma ennustettavuus, kun vähintään kuukauden välein tarkastellaan projektin edistymistä. Mikäli sprintti on liian pitkä, sprintin tavoite voi ehtiä ajan myötä hämärtyä ja samalla kompleksisuus sekä riskit kasvavat. Lyhyempien sprinttien etuna on riskien rajoittuminen pienempään aikaikkunaan, jolloin ongelmatilanteet tulevat nopeammin esiin ja suuntaa päästään korjaamaan ajoissa. Poikkeustilanteissa sprintti on myös mahdollista keskeyttää, mikäli sprintin tavoite jostain syystä todetaan tarpeettomaksi tai sen toteuttaminen ei ole kannattavaa. Ainoastaan tuoteomistaja voi keskeyttää sprintin. (Schwaber & Sutherland, 2020).

Sprintti aloitetaan aina suunnittelupalaverilla, jossa suunnitellaan ja käydään koko tiimin kesken läpi kyseisen sprintin tehtävät. Tuoteomistaja vastaa siitä, että suunnittelupalaverissa otetaan esiin tärkeimmät kehitysjonossa olevat asiat. Tarvittaessa suunnittelupalaveriin voidaan kutsua tiimin ulkopuolisia henkilöitä antamaan neuvoja tehtävien suhteen. (Schwaber & Sutherland, 2020). Suunnittelupalaverissa on syytä pohtia seuraavia kysymyksiä:

- Miksi sprintti on hyödyllinen?
- Mitä sprintin aikana ehditään tehdä?
- Miten valitut tehtävät toteutetaan?

Myös suunnittelupalaverin aika on rajattu, ja esimerkiksi kuukauden pituisen sprintin suunnittelupalaveri voi olla maksimissaan kahdeksan tunnin pituinen. (Schwaber & Sutherland, 2020).

Päivittäispalavereja pidetään nimensä mukaisesti joka päivä koko sprintin ajan. Schwaberin ja Sutherlandin (2020) mukaan päivittäispalaverissa tarkastellaan edistymistä kohti sprintin tavoitetta ja mukautetaan tarpeen mukaan kyseisen sprintin kehitysjonoa. Samalla luodaan myös suunnitelma seuraavan päivän töistä. Päivittäispalaveri on maksimissaan 15 minuutin pituinen ja siihen osallistuvat kehittäjät. Mikäli tuoteomistaja tai scrummaster itse osallistuvat aktiivisesti kyseisen sprintin tuotteen toteutukseen, osallistuvat myös he palaveriin kehittäjän roolissa. Päivittäispalaverin hyötynä on kommunikoinnin edistäminen, esteiden tunnistaminen, nopean päätöksenteon edistäminen sekä vähentynyt tarve muille palavereille. Kehittäjät saavat mukauttaa suunnitelmia myös muutoin kuin päivittäispalavereissa, ja usein kehittäjät tapaavatkin toisiinsa päivän aikana yksityiskohtaisempien keskusteluiden parissa. (Schwaber & Sutherland, 2020).

Sprintin lopussa pidettävässä sprintin katselmoinnissa tarkastellaan kyseisen sprintin lopputulosta ja sopeutetaan tarvittaessa sen pohjalta tuotteen kehitysjonoa. Katselmoinnissa tiimi esittelee sprintin tulokset sidosryhmille, mutta tämän lisäksi katselmoinnissa on tärkeää myös keskustella tulevista suuntaviivoista sen pohjalta, mitä sprintissä on saavutettu ja mitä kaikkea on muuttunut. Sprintin katselmoinnille määritelty aikaraja kuukauden pituisen sprintin tapauksessa on neljä tuntia. (Schwaber & Sutherland, 2020).

Viimeinen Scrumin tapahtumista on sprintin retrospektiivi. Sen tavoitteena on Schwaberin ja Sutherlandin (2020) mukaan löytää keinoja kehitysprosessin laadun ja tehokkuuden lisäämiseksi. Retrospektiivissä loppuneen sprintin toteutumaa peilataan suhteessa ihmisiin, prosesseihin, työkaluihin sekä yhteistyöhön. Tämän pohjalta pyritään tunnistamaan, mikä sprintin aikana meni hyvin, mitä ongelmia kohdattiin ja miten niistä mahdollisesti selvittiin. Retrospektiivin aikana tunnistetut tehokkuutta parantavat asiat pyritään ottamaan käyttöön mahdollisimman pian, tarpeen mukaan jo seuraavassa sprintissä. Retrospektiiville määritelty aikaraja on kolme tuntia. (Schwaber & Sutherland, 2020).

### 2.2.3 Scrumin artefaktit

Scrumin artefakteja ovat Schwaberin ja Sutherlandin (2020) mukaan tuotteen kehitysjono, sprintin kehitysjono sekä inkrementit. Tuotteen kehitysjonolla tarkoitetaan listaa asioista, jotka valmistuvaan tuotteeseen vaaditaan. Tyypillisesti kehitysjonon kohtia hajotetaan pienempiin ja täsmällisempiin osiin, joita voidaan toteuttaa sprinttien aikana. Sprintin kehitysjono taas koostuu tuotteen kehitysjonosta valituista kohdista, jotka on päätetty toteuttaa seuraavan sprintin aikana. (Schwaber & Sutherland, 2020).

Kaikki Scrumin artefaktit edustavat joko työmäärää tai lisäarvoa. Jokaisen artefaktin edistymistä voidaan seurata vertaamalla niitä vastaaviin tavoitteisiin: tuotteen kehitysjonon tila antaa tietoa projektissa jäljellä olevasta työmäärästä, kun kehitysjonoa verrataan projektin kokonaistavoitteisiin. Sprintin kehitysjonoa sen sijaan voidaan verrata kyseisen sprintin tavoitteisiin, ja tätä kautta voidaan laskea sprintin jäljellä oleva työmäärä. (Schwaber & Sutherland, 2020).

Inkrementillä eli tuoteversiolla tarkoitetaan kyseisen sprintin ja edellisten sprinttien aikana valmistunutta kokonaisuutta, joka vastaa jo toteutuneita kehitysjonon kohtia. Inkrementin valmiusastetta voidaan verrata "valmiin" määritelmään, jonka tulee olla kaikille projektin osapuolille yksiselitteisesti määritelty. Kun inkrementin tilaksi määritetään "valmis", tulisi kaikkien olla yhtä mieltä siitä, mitä valmiilla tarkoitetaan ja mitkä ovat laadun mittarit. Sprintin lopussa inkrementin tulee olla "valmis"-tilassa, jotta se voidaan julkaista tai esitellä sprintin katselmoinnissa. Muussa tapauksessa se palautetaan tuotteen kehitysjonoon. (Schwaber & Sutherland, 2020).

### 3 HAJAUTETTU KETTERÄ OHJELMISTOKEHITYS

Ketterä ohjelmistokehitys on ollut osa ohjelmistokehitystä 2000-luvun alusta alkaen. Myös hajautettu työskentelymalli on ollut osa IT-alaa jo pitkään, kun ulkoistamisella on tavoiteltu muun muassa kustannussäästöjä ja monipuolisempaa osaamista. Näiden kahden osa-alueen yhdistäminen, ketterän ohjelmistokehityksen ja hajautuksen, on kuitenkin koettu olevan haasteellista muun muassa kommunikoinnin ja aikaerojen vuoksi.

Ketterien menetelmien ja hajautetun ohjelmistokehityksen periaatteet ovat joiltakin osin ristiriidassa, minkä vuoksi kirjallisuudessa on käsitelty runsaasti sitä, voiko ketteriä menetelmiä ylipäätään hyödyntää hajautetussa ohjelmistokehityksessä (Jalali & Wohlin, 2010; Ramesh ym., 2006). Ketterässä manifestissa korostetaan esimerkiksi kasvokkain käytävän keskustelun merkitystä ja julistuksen takana olevissa periaatteissa mainitaan, että ”tehokkain ja toimivin tapa tiedon välittämiseksi kehitystiimille ja tiimin jäsenten kesken on kasvokkain käytävä keskustelu” (Agile Alliance, 2001). Tämä on ristiriidassa hajautettuun ohjelmistokehitykseen, jossa kasvokkain käytävälle keskustelulle ei juurikaan ole mahdollisuuksia.

Vaikka ketterien menetelmien sopivuutta hajautettuun ohjelmistokehitykseen on kyseenalaistettu, löytyy vastapainoksi myös kirjallisuutta, jossa todetaan ketterien menetelmien sopivan joiltakin toimintatavoiltaan erityisen hyvin hajautettuun ympäristöön. Esimerkiksi Paasivaara ja Lassenius (2006) esittävät ketterän, iteratiivisen mallin sopivan hajautettuun ohjelmistokehitykseen erityisesti sen vuoksi, että ketterät toimintatavat voivat auttaa hajauttamisen tuomiin haasteisiin. Jatkuva integraatio ja testaus auttavat välttämään suuria integrointiin liittyviä haasteita projektin lopussa. Myös iteratiivisen mallin mukaisesti tehtävät uusien toiminnallisuuksien toimitukset lyhyin väliajoin auttavat pysymään perillä projektin statuksesta ja projektipäällikön sekä asiakkaan on helppompaa seurata projektin etenemistä. Samalla kehittäjät saavat nopeammin palautetta tehdyistä toiminnallisuuksista. (Paasivaara & Lassenius, 2006).

Tässä luvussa käydään aluksi läpi hajautetun ohjelmistokehityksen käsitettä (alaluku 3.1), minkä jälkeen alaluvussa 3.2 siirrytään tarkastelemaan hajautettua ohjelmistokehitystä ketterän kehityksen näkökulmasta. Esimerkkinä



ketterien menetelmien käytöstä hajautetussa ympäristössä esitellään hajautetun Scrumin malli luvussa 3.3. Lopuksi alaluvussa 3.4 tarkastellaan vielä kirjallisuuden pohjalta hajautettuun ketterään ohjelmistokehitykseen liitettyjä haasteita.

### 3.1 Hajautettu ohjelmistokehitys

Ohjelmistokehitystä on tehty hajautetusti jo parin vuosikymmenen ajan. Vaikka hajauttaminen tuo mukanaan myös haasteita, koetaan hajauttamisen tuomat hyödyt, kuten kustannussäästöt ja mahdollisuus laajemman osaamisen hyödyntämiseen sekä ympärivuorokautiseen kehitykseen niin merkittäviksi, että hajauttaminen koetaan tarpeelliseksi toimintatavaksi. (Carmel & Agarwal, 2001; Matalonga ym., 2013).

Hajautetulla ohjelmistokehityksellä (Distributed Software Development, DSD) tarkoitetaan ohjelmistokehitystä, joka on jakautunut eri sijaintien välille joko siten, että erilliset tiimit toimivat eri sijainneissa tai yksittäiset tiimit työskentelevät hajautetusti eri sijainneista käsin. Hajautettu ohjelmistokehitys voi tapahtua joko yksittäisen yrityksen sisällä tai yritysten välillä, kuten esimerkiksi alihankintasuhteessa. Hajautettu ohjelmistokehitys ei ota kantaa siihen, kuinka suuresta maantieteellisestä hajautuksesta on kyse. Hajautus voikin pienimmillään olla vain kahden vierekkäisen rakennuksen välillä, joissa tiimit tai tiimien jäsenet toimivat erillään toisistaan, mutta laajimmillaan kyse voi olla ympäri maailman hajautetuista tiimeistä, jolloin hajautukseen liittyy myös muun muassa kielellisiä ja kulttuurisia eroja sekä aikaero.

Hajautetun ohjelmistokehityksen yhteydessä voidaan puhua myös globaalista ohjelmistokehityksestä (Global Software Development, GSD), joka on hajautetun ohjelmistokehityksen erikoistapaus. Globaalin ohjelmistokehityksen tapauksessa tiimi on aina jakautunut maailmanlaajuisesti eri sijainteihin. Globaalin ohjelmistokehityksen haasteita on tutkittu jo vuosien ajan, ja monissa tutkimuksissa nousee esiin samat ongelmakohdat: puutteet kommunikaatiossa, kontrollin puuttuminen, henkilöstön heikko moraalit ja epämääräiset vaatimukset (Beecham, Noll & Richardson, 2014).

Phalnikar, Deshpande ja Joshi (2009) jakavat hajautuksen kahteen kategoriaan: hajautettuun ohjelmistokehitykseen sekä yksilötasolla hajautettuun ohjelmistokehitykseen. Hajautetussa ohjelmistokehityksessä eri paikoissa sijaitsevat erilliset tiimit toimivat yhteistyössä, kun taas yksilötasolla toteutetussa hajautuksessa yksittäiset ohjelmistokehittäjät työskentelevät eri sijainneissa ja ovat yhteydessä toisiinsa verkon välityksellä. Yksilötasolla toteutetussa hajautuksessa kehittäjät voivat työskennellä esimerkiksi kotoa käsin.

### 3.2 Ketteryyden yhdistäminen hajautettuun ohjelmistokehitykseen

Ketterissä ohjelmistoprojekteissa on hyödynnetty hajautettuja tiimejä jo jonkin aikaa. Lous, Tell, Michelsen, Dittrich ja Ebdrup (2018) ovat tutkineet, miten ketterän ohjelmistoprojektin periaatteet toteutuvat ympäristössä, jossa tiimin työskentely tapahtuu hajautetusti eri toimipisteissä sekä etätyönä. Tulokset osoittavat, että hajautettuna tapahtuva ketterä kehitys vaatii erityisesti vahvaa ketterää perustaa: tiimin tulee toimia ketterien menetelmien mukaisesti ja tiimillä tulee olla ketterä, palveleva johtaja (agile servant-leader). Sekä tiimiltä että yksilöiltä edellytetään itsenäisyyttä, mikä myös samalla vaatii luottamusta siihen, että tiimit ja yksilöt suoriutuvat sovitusta tehtävistä.

Kolmantena tärkeänä asiana Lous ym. (2018) nostavat esiin kulttuurin, jossa painotetaan kokeilemisen merkitystä. Tällöin nopeasti saatu palaute auttaa näkemään epäonnistumiset oppimiskokemuksina sen sijaan, että ne koettaisiin ongelmina. Neljäntenä Lous ym. (2018) nostavat esiin jatkuvan pyrkimyksen yksinkertaisuuteen ja nopeuteen vähentämällä turhaa työtä (reducing waste).

### 3.3 Hajautettu Scrum

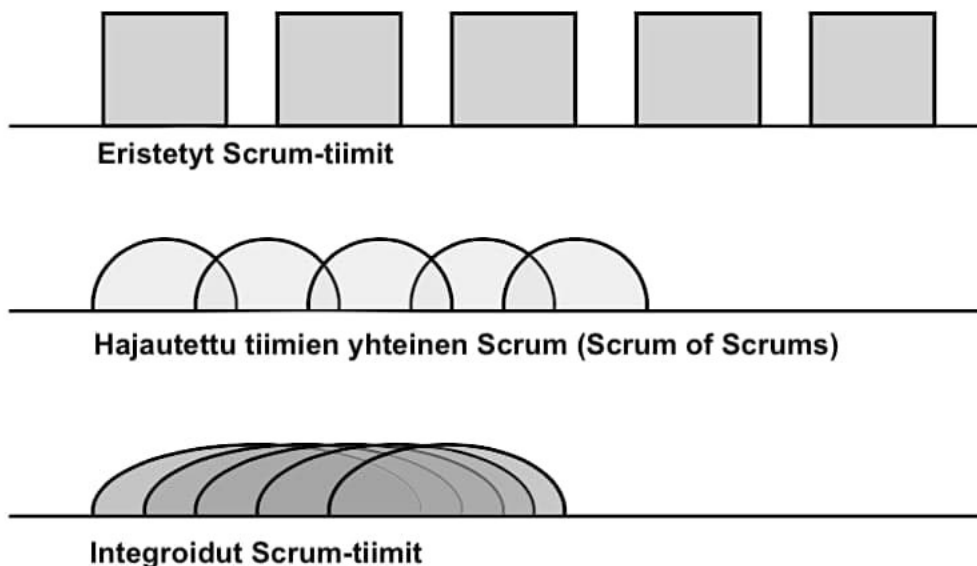
Paasivaara, Durasiewicz ja Lassenius (2009) ovat tutkineet Scrumin soveltamista hajautettuun ohjelmistokehitykseen. He jakavat tutkimuksesta saadut tulokset kahteen kategoriaan, ketteriin menetelmiin liittyviin käytänteisiin ja globaalia hajautettua ohjelmistokehitystä tukeviin käytänteisiin. Ketteriin menetelmiin liittyvistä käytänteistä erityisesti Scrumin päiväpalaverit koettiin tärkeiksi hajautetun työn kannalta, sillä ne antoivat osallistujille mahdollisuuden saada päivittäin projektin ja työtehtävien kannalta tarpeellista tietoa. Kokonaisuudessaan Scrumin käyttäminen hajautetussa ohjelmistoprojektissa koettiin toimivaksi ratkaisuksi, sillä Scrumiin liittyvät käytänteet, kuten päiväpalaverit, viikkopalaverit, retrospektiivit ja sprintin suunnittelu, tarjosivat jo itsessään tavan kommunikoida avoimesti ja usein, mikä lisäsi luottamusta hajautetussa ympäristössä toimivien kehittäjien kesken. Tiimin jäsenet keskustelivat projektiin liittyvistä ongelmista avoimemmin kuin ennen Scrumiin siirtymistä, ja samalla Scrumin kautta "pakotettu" tiivis kommunikaatio lisäsi myös tiimin jäsenten keskinäistä epämuodollista kommunikaatiota, kun keskustelun kynnys palaverien kautta oli madaltunut. Scrumin nopeat iteraatiot ja päivittäiset palaverit tarjosivat mahdollisuuden tarkastella tiiviimmin myös muiden toimipisteiden työskentelyä, mikä auttoi huomaamaan vaatimuksiin liittyneitä väärinymmärryksiä ja mahdollisia ongelmia. (Paasivaara, Durasiewicz & Lassenius 2008; Paasivaara ym., 2009).

Myös Alzoubi, Gill ja Al-Ani (2016) toteavat tutkimuksessaan, että Scrumin käytänteet erityisesti globaalisti hajautetussa ketterässä ohjelmistokehityksessä voivat edistää kommunikaatiota ja yhteistyötä tiimien välillä. Scrumin kantavana ideana on lisätä ketterään ohjelmistokehitysprosessiin selkeyttä ja läpinäkyvyyttä, mikä on hajautetussa ohjelmistokehityksessä tärkeässä osassa, jotta kaikki tiimissä pysyvät perillä vaatimuksista ja tavoitteista (Phalnikar ym., 2009; Sutherland, Viktorov, Blount & Puntikov, 2007).

Hajautetussa ohjelmistokehityksessä erääksi haasteeksi on koettu epäselvyys projektiin liittyvien henkilöiden ja tiimin jäsenten rooleista ja vastuista. Scrumin yksiselitteisesti määritellyt roolit, kuten tuoteomistaja, scrummaster ja projektitiimin jäsenet, sekä kullekin roolille määritellyt vastuut voivatkin vähentää merkittävästi rooleihin ja vastuisiin liittyviä epäselvyyksiä. (Beecham ym., 2014).

Hajautetussa ohjelmistokehityksessä haasteeksi nostetaan usein myös vaatimusten epämääräisyys tai jopa puuttuminen. Scrumin lyhyet sprintit ja jokaisen sprintin alussa tehtävä tuotteen kehitysjonon tarkastelu sekä sprintin lopussa asiakkaan kanssa toteutettava katselmus auttavat erityisesti pysymään ajan tasalla vaatimuksista sekä mahdollisista haasteista. (Beecham ym., 2014).

Sutherland ym. (2007) ovat tutkineet Scrumin hyödyntämistä hajautetuissa ohjelmistoprojekteissa ja luoneet sen pohjalta kolme mallia Scrumin toteuttamiseksi tiimien yhteistyön tason pohjalta (kuvio 1): eristetyt Scrum-tiimit, hajautettu tiimien yhteinen Scrum (Scrum of Scrums) sekä integroidut Scrum-tiimit.



KUVIO 1 Hajautetun Scrumin mallit Sutherlandin ym. (2007) mukaisesti

Sutherlandin ym. (2007) mukaan eristetyt Scrum-tiimit toimivat itsenäisesti ja ovat maantieteellisesti erillään toisistaan, eivätkä tiimit välttämättä käytä Scrum-prosessia. Myös hajautetussa tiimien yhteisessä Scrumissa tiimit ovat

maantieteellisesti erillään, mutta niitä yhdistää säännöllisesti pidettävä usean tiimin yhteinen Scrum (Scrum of Scrums), jossa scrummasterit tapaavat. Tämä toimintatapa tukee kommunikaatiota ja yhteistyötä ja sopii erityisesti tilanteisiin, joissa ketterä kehitys on uusi toimintatapa. (Sutherland ym., 2007).

Integroidut Scrum-tiimit toimivat tiiviisti yhteistyössä keskenään ja myös yksittäisten tiimien jäsenet työskentelevät maantieteellisesti hajautetusti. Tässä mallissa päivittäispalaverien tarve korostuu, jotta kommunikaatio pysyy riittäväällä tasolla ja eri kulttuureihin liittyvien eroavaisuuksien aiheuttamia haasteita saadaan vähennettyä. Vaikka kommunikaatiosta on erityisesti pidettävä huolta integroitujen Scrum-tiimien tapauksessa, luo tämä lähestymistapa huolellisesti toteutettuna läpinäkyvyyttä toimipisteiden välille ja lähentää tiimin jäseniä. (Sutherland ym., 2007).

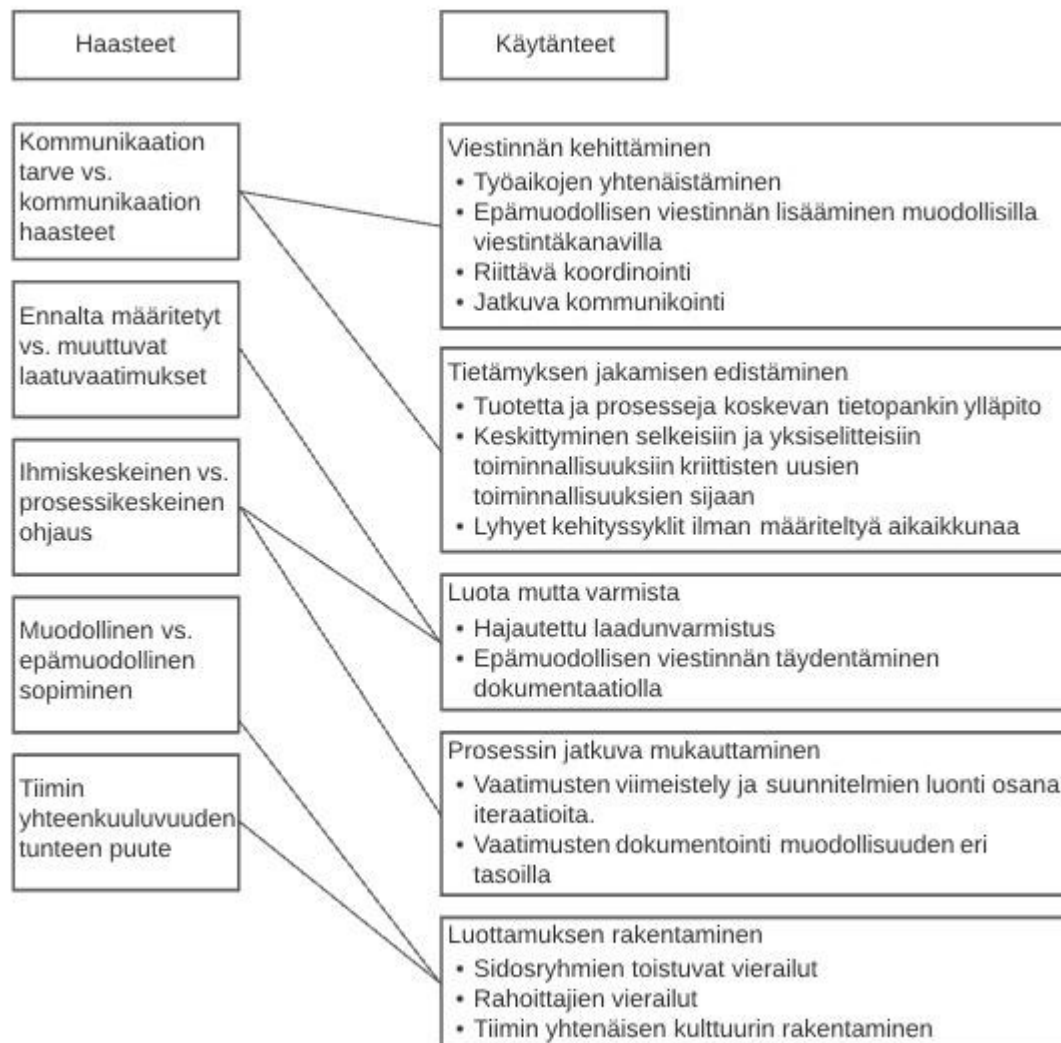
Myös XP:ssä (Extreme Programming) on toimintatapoja, joista voi olla hyötyä hajautetussa ympäristössä toimiessa. Holmström, Fitzgerald, Ågerfalk ja Conchúir (2006) nostavat tutkimustuloksissaan esiin muun muassa pariohjelmoinnin hyödyn tapauksissa, joissa parin osapuolet työskentelevät eri sijainneista. Pariohjelmoinnin todettiin tukevan kehittäjien sitoutumista sekä vähentävän ajallista etäisyyttä, kun ohjelmoijaparit pyrkivät luomaan työaikataulunsa siten, että he olisivat mahdollisimman yhteneväisiin aikoihin töissä, vaikka toimisivatkin eri aikavyöhykkeillä (Holmström ym., 2006).

### **3.4 Hajautetun ketterän ohjelmistokehityksen haasteet**

Monet hajautettuun ketterään kehitykseen liittyvistä haasteista liittyvät hajautettuun ohjelmistokehitykseen ja etäisyyteen, eivätkä niinkään itse ketterän kehityksen toimintatapoihin tai arvoihin (Matalonga ym., 2013). Berczukin (2007) mukaan hajautus korostaa ja laajentaa jo olemassa olevia prosessin ongelmia eikä ole itsessään ongelmien aiheuttaja. Etäisyyden aiheuttamiin rajoituksiin liittyen hajautetussa työskentelyssä haasteita aiheutuu myös kommunikaation osalta.

#### **3.4.1 Hajautetun ketterän ohjelmistokehityksen haasteet Rameshin ym. (2006) mukaan**

Ramesh ym. (2006) ovat tutkineet hajautettua ketterää ohjelmistokehitystä ja nostavat tutkimuksessaan esiin viisi tyypillistä siihen liittyvää haastetta, sekä käytänteitä niiden ratkaisemiseksi (kuvio 2). Haasteet ovat ristiriita kommunikaation tarpeen ja haasteiden välillä, ristiriita ihmiskeskeisen ja prosessikeskeisen ohjauksen välillä, ristiriita kiinnitettyjen ja muuttuvien laatuvaatimusten välillä, tiimin yhteenkuuluvuuden tunne sekä ristiriita muodollisen ja epämuodollisen sopimisen välillä. (Ramesh ym., 2006).



KUVIO 2 Hajautetun ketterän ohjelmistokehityksen haasteet (Ramesh ym., 2006, s. 44)

Rameshin ym. (2006) mukaan ketterissä menetelmissä korostuu tyypillisesti epämuodollisen kommunikaation osuus, kun taas vastaavasti muodollista viestintää on hyvin vähän. Hajautetuissa projekteissa sen sijaan kommunikaatio aiheuttaa haasteita erityisesti aikaeron, väärinymmärrysten, keskustelun aikaansaamisen sekä ylipäätään tavallista niukemman viestinnän vuoksi. Kun sekä hajautettujen projektien kommunikaation haasteet että ketterille menetelmille tyypilliset kommunikointitavat yhdistetään, on lopputuloksena hajautetuissa ketterissä ohjelmistoprojekteissa havaittavissa ristiriita kommunikaation tarpeen ja sen haasteiden välillä (Ramesh ym., 2006). Hajautetussa ympäristössä pitäisi löytää sopiva tasapaino ketteryyden korostaman epämuodollisen viestinnän ja hajautetussa ympäristössä tarvittavan muodollisen viestinnän välille.

Ramesh ym. (2006) ehdottavat kommunikaation aiheuttamaan ristiriitaan ratkaisuksi työaikojen yhtenäistämistä ja epämuodollisen kommunikaation lisäämistä käyttäen muodollisia viestintäkanavia. Kommunikointiin liittyy myös tietämyksen jakaminen, mitä voi edistää esimerkiksi ylläpitämällä ajantasaista tietopankkia tuotetta ja prosessia koskevista asioista. Ramesh ym. (2006) suositt

televat myös keskittymään aluksi selkeiden ja ymmärrettävien toiminnallisuuksien kehittämiseen uusien ja kriittisten sijaan sekä pitämään kehittämisen lyhyinä sykleinä, joille ei kuitenkaan ole määritelty tarkkaa aikaikkunaa.

Rameshin ym. (2006) mukaan kontrollin puute on hajautetun ohjelmistokehityksen osalta tunnistettu tyypillinen haaste. Prosesseja ja laatua on haastavaa hallita ympäristössä, jossa tiimit ovat hajautettuja, mistä johtuen hajautetuissa projekteissa suositaan ennalta määriteltyjä laatuvaatimuksia sekä muodollisia prosesseja. Sen sijaan ketterissä malleissa korostuu ihmislähtöisyys sekä prosessien pitäminen mahdollisimman keveinä samalla kun tiimin jäsenten osaamiseen luotetaan vahvasti. Myös laatuun liittyviä vaatimuksia määritellään ketterissä lähestymistavoissa yhteistyössä asiakkaan kanssa jatkuvasti projektin aikana. Ketterän ja hajautetun kehityksen yhdistelmässä kontrollin puute kiteytyykin Rameshin ym. (2006) mukaan erityisesti ristiriidoiksi ihmiskeskeisen ja prosessikeskeisen ohjauksen välillä sekä kiinnitettyjen ja muuttuvien laatuvaatimusten välillä. Näihin haasteisiin tutkimuksessa ehdotetaan ratkaisuksi hajautettua laadunvarmistusta ja dokumentaation luontia epämuodollisen informaation tueksi, sekä prosessin jatkuvaa mukauttamista ja sen yhteydessä erityisesti iteraatioiden huolellista suunnittelua sekä vaatimusten dokumentointia.

Hajautetussa ohjelmistokehityksessä tiimin yhteenkuuluvuuden tunne on usein heikompi kuin tilanteessa, jossa tiimi toimii samassa tilassa, mikä tuo erityistä haastetta ketterään ohjelmistokehitykseen, jossa tiimin yhteistyön tarve, luottamus ja sitoutuminen korostuvat. Tiimin yhteenkuuluvuuden tunnetta voidaan pyrkiä kasvattamaan rakentamalla luottamusta tiimin jäsenten välille. Luottamuksen rakentaminen vaatii tiimin yhtenäisen kulttuurin luomista. (Ramesh ym., 2006).

Viidenneksi haasteeksi hajautettuun ketterään kehitykseen liittyen Ramesh ym. (2006) nostavat ristiriidan muodollisen ja epämuodollisen sopimisen välillä, mikä myös pohjautuu hajautetuille projekteille tyypilliseen luottamuksen puutteeseen. Ketterissä menetelmissä suositaan löyhästi ja epämuodollisesti määriteltyjä sopimuksia, kun taas hajautetussa ympäristössä luotetaan tarkasti määriteltyihin tavoitteisiin ja yksityiskohtaiseen vaatimusmäärittelyyn. Haasteena onkin löytää sopimusten luonnille sopiva muodollisuuden taso. Myös tähän Ramesh ym. (2006) ehdottavat ratkaisuksi luottamuksen rakentamista esimerkiksi sidosryhmien sekä rahoittajien paikan päällä vierailuja lisäämällä.

### **3.4.2 Hajautetun ketterän ohjelmistokehityksen haasteet muissa tutkimuksissa**

Rameshin ym. (2006) tutkimuksen lisäksi kirjallisuudesta löytyy myös muita hajautettujen ketterien ohjelmistoprojektien haasteisiin keskittyneitä tutkimuksia. Monet näistä on toteutettu tapaustutkimuksina, mutta esimerkiksi Kajko-Mattsson, Azizyan ja Magarian (2010) ovat koonneet tapaustutkimusten pohjalta oman 13 kohdan listauksensa haasteista, joita hajautetuissa ketterissä ohjelmistoprojekteissa tyypillisesti kohdataan. Nämä haasteet on kerätty aihealueittain taulukkoon 1.

TAULUKKO 1 Hajautetun ketterän ohjelmistokehityksen haasteet Kajko-Mattssonin ym. (2010) mukaan

<b>Aihealue</b>	<b>Haaste</b>
Kulttuuri	Käsitys vastuiden jakautumisesta Käsitys välittömyydestä ja rehellisyydestä Käsitys auktoriteetista Kielelliset haasteet
Aikaerot	Kansalliset vapaapäivät ja loma-ajat Työaikojen yhtenäistäminen ja suunnittelu
Kommunikaatio	Yhteistyön tehokkuus Lisääntynyt dokumentaation määrä Etäkokousten tehottomuus
Luottamus	Tiimin yhteenkuuluvuuden tunne
Asiakasyhteistyö	Yhteydenpito asiakkaaseen tai asiakkaan edustajaan
Koulutus	Osaamisen eroavaisuudet
Käytettävät teknologiat	Tekniset ongelmat

Kajko-Mattssonin ym. (2010) mukaan kulttuuriin liittyvien haasteiden taustalla on se, että eri kulttuureihin kuuluu muun muassa erilaisia arvoja, asenteita, hierarkioita, yleisiä uskomuksia ja aikakäsityksiä. Nämä erilaiset käsitykset voivat aiheuttaa haasteita ja väärinymmärryksiä tilanteissa, joissa eri kulttuuritaustoista tulevat henkilöt työskentelevät keskenään. Tästä johtuen henkilöiden tekemät oletukset projektihenkilöiden vastuiden jakautumisesta, käsitys välittömyyden ja rehellisyyden tarpeesta sekä auktoriteeteista kuten myös kielelliset haasteet on erityisen tärkeää osata ottaa huomioon hajautetuissa projekteissa.

Aikaeroihin liittyviksi haasteiksi Kajko-Mattsson ym. (2010) lukevat tarpeen yhtenäistää eri aikavyöhykkeillä toimivien henkilöiden työaikoja sekä kansallisten vapaapäivien sekä loma-aikojen huomioon ottamisen. Tutkimuksessa todettiin aikaeroihin liittyvien haasteiden aiheuttavan tiimin jäsenissä väsymystä sekä turhautumista, minkä vuoksi on erittäin tärkeää pyrkiä löytämään muun muassa kaikille sopivia aikoja palaverien pitämiseen.

Kommunikaatio on monissa tutkimuksissa nostettu yhdeksi merkittävimmistä haasteista hajautetussa ympäristössä työskennellessä, ja myös Kajko-Mattsson ym. (2010) ovat eritelleet tähän aihealueeseen liittyviä ongelmakohtia. Tutkimuksen mukaan tehokas kommunikaatio ilman ketterille menetelmille tyypillistä kasvokkain tapahtuvaa kommunikaatiota on haastavaa, mutta tämän lisäksi hajautetuissa projekteissa myös dokumentaation määrän merkittävä lisääntyminen on ristiriidassa ketterien menetelmien korostaman vähäisen dokumentaation tarpeen kanssa. Vaikka hajautetuissa projekteissa dokumentaatiota tarvitaan enemmän rajallisten kommunikointimahdollisuuksien vuoksi, tulisi Kajko-Mattssonin ym. (2010) mukaan dokumentaatiota pyrkiä vähentä-

mään sellaiseen määrään, että se on riittävää kommunikoinnin ylläpitämiseksi. Kolmantena haasteena kommunikointiin liittyen Kajko-Mattsson ym. (2010) nostavat esiin etäkokouksiin ja niiden tehokkuuteen liittyvät haasteet. Usein etäkokousten seuraaminen on haastavampaa liittyen teknisiin haasteisiin, mutta myös tiimin jäsenten osallistuminen ja aktiivisuus kokouksissa on heikompaa.

Luottamukseen liittyvät haasteet Kajko-Mattsson ym. (2010) tiivistävät tiimin yhteenkuuluvuuden tunteen vähäisyyteen, joka pohjimmiltaan on seurausta vähäisestä kasvokkain tapahtuvasta viestinnästä. Hajautetuissa tiimeissä onkin tarpeen kiinnittää erityistä huomiota tiimin jäsenten välisten suhteiden luomiseen sekä tehokkaaseen yhteistyöhön.

Ketterissä menetelmissä korostuu myös yhteistyö asiakkaan kanssa, mikä vaatii tiimin lisäksi asiakkaan puolelta aktiivisuutta yhteydenpitoon. Erityisen haastavaksi asiakkaan tavoitettavuuden tekee se, ettei esimerkiksi ulkoistetulla projektitiimillä ole välttämättä suoraa kommunikointimahdollisuutta asiakkaan kanssa. Kajko-Mattsson ym. (2010) nimeävätkin yhdeksi haasteeksi asiakasyhteistyön osalta asiakkaan tai asiakkaan edustajan tavoitettavuuden.

Koulutuksen osalta haasteita voi Kajko-Mattssonin ym. (2010) mukaan aiheuttaa eroavaisuudet osaamisessa, mikä laskee tehokkuutta sekä vaikeuttaa tiimityötä. Teknisen osaamisen erojen lisäksi tiimeissä voi ilmetä esimerkiksi erilaisia käsityksiä ketterien menetelmien käytänteistä, jolloin eri henkilöt olettavat toisiltaan erilaisia työskentelytapoja. Viimeisenä osa-alueena Kajko-Mattsson ym. (2010) nostavat esiin tekniset haasteet, kuten infrastruktuurin, eri teknologioiden saatavuuden sekä erilaiset standardit, mikä osaltaan tuo haasteita tiimien ja henkilöiden väliseen yhteistyöhön.

Kajko-Mattssonin ym. (2010) listaus hajautettujen ketterien projektien ongelmakohdista on laaja, mutta monet listauksessa esitellyistä haasteista ja ongelmista ovat sellaisia, joita voi esiintyä myös sellaisissa projekteissa, joita ei ole toteutettu hajautetusti. Esimerkiksi Ramesh ym. (2006) keskittyvät tutkimuksessaan erityisesti sellaisiin ongelmiin, jotka juontuvat hajautetun projektin ja ketteryyden yhdistelmästä, kun taas Kajko-Mattssonin ym. (2010) listaukseen on huomioitu myös sellaiset tekijät, jotka ovat ainoastaan hajautuksen mukanaan tuomia haasteita.

Myös Lee ja Yong (2010) ovat tutkineet hajautettuun ketterään ohjelmistokehitykseen liittyviä haasteita sekä toimintatapoja haasteiden vähentämiseksi. Lee ja Yong (2010) jakavat globaalien ohjelmistokehitystiimien kohtaamat haasteet Rameshin ym. (2006) jaottelun mukaisesti kommunikaatioon, kontrolliin sekä luottamukseen liittyviin haasteisiin. Ratkaisuksi globaalisti hajautettujen projektien aiheuttamiin haasteisiin Lee ja Yong (2010) ehdottavat hajautetun Scrum of Scrums -mallin käyttöä, jonka mukaisesti globaalit tiimit jakautuvat useisiin pienempiin tiimeihin, joilla kaikilla on oma scrummaster. Kullekin paikalliselle tiimille Lee ja Yong (2010) suosittelivat omaa scrummasteria, joka taas toimii tiiviissä yhteistyössä muiden lokaalien tiimien scrummastereiden kanssa. Scrummastereiden käyttö tiimeissä vähentää Leen ja Yongin (2010) mukaan yksittäisen lokaalin projektipäällikön vastuuta lukuisista eri toiminnoista ja helpottaa näin ollen projektin hallintaan liittyvää työtaakkaa.



Scrum of Scrums on Leen ja Yongin (2010) mukaan hyödyllinen korkeamman tason koordinoinnin ylläpitämiseen, kun käytössä on useita erillisiä tuotteen kehitysjojoja. Kullakin tiimillä tulisi olla mahdollisuus pitää omat palaverinsa oman aikavyöhykkeen mukaisina työaikoina ja koko tiimille on tärkeää tarjota projektin alussa koulutusta liittyen ketterien menetelmien käyttöön. Luottamuksen lisäämiseksi Lee ja Yong (2010) ehdottavat hyödyntämään kasvokkain tapahtuvaa keskustelua niin usein kuin mahdollista, mukauttamaan Scrumin prosesseja sekä antamaan arvoa ihmisille ja heidän panokselleen.

### 3.4.3 Etäisyys

Suurimmaksi haasteeksi hajautetun ketterän ohjelmistokehityksen tutkimuksissa nousee etäisyyteen liittyvät haasteet. Holmström ym. (2006) jakavat etäisyyden kolmeen osa-alueeseen: ajallinen etäisyys, maantieteellinen etäisyys ja sosiokulttuurinen etäisyys. Matalonga ym. (2013) käyttävät samaa jaottelua, mutta lisäävät siihen myös teknisen etäisyyden.

Holmströmin ym. (2006) mukaan maantieteellinen etäisyys vaikeuttaa kasvokkain tapahtuvaa kommunikaatiota ja lisää kustannuksia, mikäli eri paikoissa työskentelevät ihmiset joutuvat liikkumaan enemmän toimipisteiden välillä. Epämuodollisen keskustelun puute voi vaikuttaa työtehtävien kannalta tarpeellisen tiedon määrään. Esimerkiksi kahvipöytäkeskusteluissa nousee usein esiin erilaisia näkemyksiä ja muuta tietoa, jota ei välttämättä palavereissa huomata nostaa esiin. Ajallinen etäisyys sen sijaan vaikeuttaa samanaikaista kommunikointia. Aikaerojen vuoksi tiimin jäsenet ovat eri aikoihin töissä, mikä hidastaa viestintää. Sosiokulttuurinen etäisyys taas lisää riskiä väärinkäsityksille. Epäjohtamukaisuus toimintatavoissa voi vaikeuttaa koordinointia ja yhteistyötä ja erilaiset näkemykset auktoriteetista ja hierarkiasta voivat asettaa haasteita eri kulttuurien välille. (Holmström ym., 2006).

Olson & Olson (2000) tarkastelevat tutkimuksessaan hajautetun työn tehokkuutta erityisesti sosioteknisestä näkökulmasta. Erityisesti haasteiksi paikan päällä tapahtuvaan työskentelyyn verrattuna nousivat yhteisen kontekstin puute (common ground) sekä työtehtävien kytkeytyminen toisiinsa (coupling of work). Mitä rutiinomaisempia työtehtävät ovat, sen helpompia ne ovat suorittaa myös hajautetussa ympäristössä. Sen sijaan sellaiset tehtävät, jotka kytkeytyvät toisiinsa ja vaativat muun muassa paljon suunnittelua sekä useamman henkilön osaamista, koettiin haastaviksi toteuttaa ilman kasvokkain tapahtuvaa kommunikointia. (Olson & Olson, 2000).

## 4 PROJEKTIPÄÄLLIKÖN KOMPETENSSIT

Projektipäällikkö on tärkeässä osassa projektin läpiviennissä sekä projektitiimin johtamisessa, ja usein projektipäällikön toimintaa peilataan suoraan projektin onnistumiseen. Projektipäällikön kompetenssilla, taidoilla ja toimintatavoilla onkin iso merkitys projektin kannalta. Kompetenssilla tarkoitetaan tietoa, asennetta ja taitoja, joita vaaditaan työtehtävästä suoriutumiseen. Kompetenssi voidaan nähdä myös kykynä soveltaa relevanttia tietoa ja taitoa työtehtävästä suoriutumiseen. (Dillon & Taylor 2015).

Tässä luvussa käydään läpi erityisesti IT-alan projektipäällikölle tarpeellisia kompetensseja. Alaluvussa 4.1 käsitellään projektipäälliköiden koulutusta ja sertifikaatteja, ja alaluvussa 4.2 keskitytään IT-alan projektipäällikön kompetensseihin tarkastellen niitä kolmesta eri näkökulmasta: mitä taitoja projektipäälliköt itse pitävät tärkeinä, mitä projektitiimit haluavat projektipäälliköiden osaavan ja millaista osaamista johtoporras arvostaa palkatessaan projektipäälliköitä. Alaluvussa 4.3 keskitytään Keilin, Leen ja Dengin (2013) laajaan listaukseen IT-alan projektipäällikön kompetensseista. Kyseistä Keilin ym. (2013) listausta käytetään kattavuutensa vuoksi perustana myös luvun 5 analyysissä, jossa tarkastellaan projektipäällikön tärkeimpiä osaamisalueita hajautettujen ketterien ohjelmistoprojektien yhteydessä.

### 4.1 Projektipäälliköiden koulutus ja sertifikaatit

Projektipäälliköiden osaamisen kehittämiseksi sekä tehokasta projektinhallintaa varten on kehitetty lukuisia koulutuksia ja sertifikaatteja. Maailmanlaajuisesti suosituimmat projektinhallinnan menetelmät ovat PMBOK (Project Management Body of Knowledge) sekä PRINCE2 (Projects IN Controlled Environment Version 2) (Karaman & Kurt, 2015). PRINCE2 koostuu seitsemästä teemasta, jotka ovat liiketoiminnallinen oikeutus, organisaatio, laatu, suunnitelmat, riski, muutos ja edistyminen, kun taas PMBOKin perustana on kymmenen tietämysaluetta: integraation hallinta, laajuuden hallinta, ajankäytön hallinta, kustan-

nusten hallinta, laadun hallinta, henkilöstön hallinta, viestinnän hallinta, riskien hallinta, hankintojen hallinta sekä sidosryhmien hallinta (Karaman & Kurt, 2015).

Ketterän lähestymistavan tultua merkittäväksi osaksi ohjelmistokehitystä on myös projektinhallinnan saralla alettu kiinnittää enemmän huomiota ketteryyteen osana projektinhallinnan menetelmiä. Erityisesti on ollut tarvetta skaalata ketteriä malleja, jotta myös laajoissa projekteissa ketterien menetelmien toimintatapoja voitaisiin käyttää ongelmitta. Skaalautuminen ei kuitenkaan ole yksinkertaista, sillä laajat projektit ovat usein maailmanlaajuisesti hajautettuja ja niissä voi olla useita keskenään yhteistyötä tekeviä tiimejä. Erityisesti laajoja projekteja varten on kehitetty muun muassa SAFe (Scaled Agile Framework), joka on kattava kokonaisuus rooleja ja toimintatapoja suurien organisaatioiden tarpeisiin. (Ebert & Paasivaara, 2017).

## 4.2 IT-alan projektipäällikön kompetenssit

Napier, Keil ja Tan (2009) ovat jakaneet projektin onnistumisen kannalta tärkeät projektipäällikön taidot yhdeksään taitokategoriaan. Nämä IT-alan projektipäälliköitä haastattelemalla saadut kategoriat ovat suunnittelu ja kontrollointi, yleinen johtaminen, johtajuus, kommunikointi, tiimin kehittäminen, asiakkaan hallinta, järjestelmäkehitys, ongelmanratkaisu sekä rehellisyys. El-Sabaa (2001) sen sijaan jakaa tutkimuksessaan projektipäällikön osaamisen kolmeen kategoriaan: ihmistaitoihin, organisatorisiin taitoihin sekä teknisiin taitoihin. Tutkimuksen havaintojen perusteella projektipäälliköt itse kokevat ihmistaidot kaikkein tärkeimmäksi osa-alueeksi, kun taas teknisten taitojen merkitys koettiin huomattavasti vähäisemmäksi (El-Sabaa, 2001). El-Sabaan (2001) tutkimus ei kuitenkaan keskity ainoastaan IT-alan projektipäälliköiden taitoihin, vaan pohjautuu yleisesti eri alojen projektipäälliköihin.

Medina ja Francis (2015) tutkivat IT-alan projektipäälliköiden kompetensseja siltä kannalta, mitä ohjelmistokehitystiimin jäsenet pitävät projektipäällikön tärkeimpinä taitoina. Kyseisen tutkimuksen mukaan tiimin jäsenet kokevat projektipäällikön merkittävimiksi taidoiksi tärkeysjärjestyksessä seuraavat:

1. Työn organisointi
2. Yhteistyökyky
3. Ihmistaidot
4. Kommunikointi
5. Projektin ulkoinen viestintä
6. Tekninen osaaminen
7. Stressinsietokyky.

Medinan ja Francisin (2015) tuloksista on nähtävissä, että myös kehitystiimi arvostaa korkealle projektipäällikön osaamista henkilöiden johtamisen saralta.

IT-alan johdon näkökulmaa projektipäällikölle tarpeellisiin kompetensseihin on esitelty Stevensonin ja Starkweatherin (2010) tutkimuksessa. Tutkimuksessa selvitettiin, mitä ominaisuuksia johto pitää tärkeinä kriteereinä palkataan uusia projektipäälliköitä. Myös tästä näkökulmasta projektipäällikön tärkeimmiksi kompetensseiksi nousivat ihmisten johtaminen, kommunikointitaidot, kirjallinen ja suullinen viestintä sekä kyky hallita muutosta ja epävarmuutta. Esimerkiksi työkokemusta, koulutusta tai teknistä osaamista ei arvostettu niin tärkeinä ominaisuuksina kuin edellä mainittuja. (Stevenson & Starkweather, 2010).

IT-alan projektipäällikön kompetensseja on tutkittu myös projektin onnistumisen näkökulmasta. Silva de Araújo ja Pedron (2015) haastattelivat IT-alan ammattilaisia ja pyrkivät selvittämään, mitkä pidetään projektipäällikön tärkeimpinä ominaisuuksina. Tärkeimmiksi ominaisuuksiksi projektin onnistumisen kannalta arvioitiin tiimin johtaminen, kohdealueen tuntemus, kommunikaatio, projektin johtaminen sekä ihmistaidot (Silva de Araújo & Pedron, 2015).

### **4.3 IT-alan projektipäällikön taidot Keilin ym. (2013) mukaan**

Keil ym. (2013) ovat koonneet projektipäälliköitä haastattelemalla listan IT-alan projektipäällikölle tarpeellisista taidoista. Kattava listaus tutkimuksessa löytyneistä taidoista on lajiteltu taulukkoon 2 kategorioittain (taulukko 2). Taidot jakautuvat kymmeneen kategoriaan: tiimijohtaminen, liiketoiminta-alueen tuntemus, kommunikaatio, ihmistaidot, tekninen osaaminen, projektin johtaminen, henkilökohtaiset ominaisuudet, organisointi, ongelmanratkaisu ja ammatillisuus (Keil ym. 2013).

Edellä mainituista kymmenestä kategoriasta kommunikaatio, tiimijohtaminen, projektin johtaminen sekä ihmistaidot sisältävät taitoja, jotka Keilin ym (2013) tutkimuksessa arvostettiin tärkeimmiksi IT-alan projektipäällikön työssä. Yksittäisistä taidoista tärkeimmiksi koettiin johtaminen, suullinen viestintä, projektin laajuuden hallinta, kuuntelutaito ja projektin suunnittelu. Tärkeimmät yksittäiset taidot on listattu tärkeysjärjestyksessä taulukkoon 3. Huomionarvoista on, että yksikään tärkeimmiksi valituista taidoista ei korosta esimerkiksi teknisten taitojen merkitystä, vaan merkittävämmäksi koettiin ihmisten väliin vuorovaikutukseen ja projektinhallintaan liittyvät ominaisuudet.

TAULUKKO 2 IT-alan projektipäällikön taidot kategorioittain (Keil ym., 2013, s. 402)

Kategoria	Kuvaus	Taidot
Tiimijohtaminen	Tiimijohtamiseen kuuluvat taidot, joita tarvitaan projektin jäsenten johtamiseen sekä tiimien välillä että tiimien sisällä. Johtamisen lisäksi projektipäällikön tulee motivoida ja voimaannuttaa tiimin jäseniä.	Kyky motivoida tiimin jäseniä Seuraajien perehdyttäminen Saavutusten huomiointi Johtajuus Yhteistyö Kyky rakentaa yhteyksiä eri tiimien välille Etätiimien hallinta
Liiketoiminta-alueen tuntemus	Taidot, joita projektipäällikkö tarvitsee liikekumppanien kanssa toimiessa. Kyky ymmärtää projektin kokonaiskuva myös sidosryhmien kannalta.	Kyky ymmärtää liiketoiminta-aluetta Kyky tunnistaa sidosryhmät Liiketoimintataidot Lopputuotteen tuntemus Dokumentointitaidot Strateginen ajattelu
Kommunikaatio	Kyky kommunikoida tehokkaasti kaikkien projektiin kuuluvien tahojen kanssa. Myös kyky kuunnella on tärkeä osa kommunikaatiota, sillä se auttaa ymmärtämään projektin osapuolien odotuksia sekä tunnistamaan mahdollisia haasteita.	Suullinen viestintä Kirjallinen viestintä Kuuntelutaito Kyky argumentoida
Ihmistaidot	Ihmistaitoja tarvitaan pitämään yllä suhteita projektiin kuuluvien tahojen kanssa. Tähän kuuluu eri sidosryhmien näkemysten ymmärtäminen esimerkiksi konfliktitilanteissa.	Suhteiden rakentaminen Ihmisten toiminnan ymmärtäminen Konfliktien hallinta Vuorovaikutustaidot Neuvottelutaidot
Tekninen osaaminen	Teknisiin taitoihin sisältyy alalla tarvittava tekninen tietämys. Tähän osa-alueeseen kuuluu myös tietämys ohjelmistokehityksen menetelmistä, prosesseista ja tekniikoista.	Tekniset taidot Kehittämismenetelmien ymmärrys

(jatkuu)

## Taulukko 2 (jatkuu)

Projektin johtaminen	Projektin johtamiseen kuuluu kyky suunnitella, monitoroida ja hallita projektia sekä sen laajuutta, resursseja ja riskejä siten, että projekti toteutuu ajallaan ja budjetissa. Tähän kategoriaan kuuluu myös tietämys projektinhallinnan työkaluista ja tekniikoista.	Projektin laajuuden hallinta Projektin suunnittelu Ajankäytön hallinta Resurssien hallinta Projektin lopetus Projektijohtamisen työkalut Projektin toimitus Kustannustenhallinta Riskien hallinta
Henkilökohtaiset ominaisuudet	Projektipäälliköllä voi olla henkilökohtaisia ominaisuuksia, jotka ovat hyödyllisiä projektin läpiviennin kannalta. Henkilöllä voi olla tällaisia ominaisuuksia luonnostaan, mutta niitä voi myös kehittää.	Huumorintaju Kärsivällisyys Stressinsietokyky Kyky tavoitella yhteisymmärrystä Sinnikkyys Yhteistyötaidot Kyky huomioida yksityiskohtia
Organisointi	Organisointiin kuuluu projektin toiminnan ja resurssien koordinointi. Organisointia tarvitaan tehtävien riippuvuuksien ja määräaikojen hallinnassa.	Organisointikyky Kyky tehdä asioita samanaikaisesti, multitasking
Ongelmanratkaisu	Kyky tunnistaa, analysoida ja ratkaista projektiin liittyviä ongelmia ja haasteita.	Analyttiset taidot Tiedonhakutaidot
Ammatillisuus	Ammatillisuudella tarkoitetaan projektipäällikön arvoja ja ominaisuuksia, jotka viestivät tinkimättömyydestä ja sitoutumisesta laatuun.	Vakuuttavuus Sitoutuminen Laatukeskeisyys Ammattitaito

TAULUKKO 3 Projektipäällikön keskeisimmät taidot (Keil ym., 2013, s. 403)

<b>Taito (tärkeysjärjestyksessä)</b>	<b>Kategoria</b>
1. Johtajuus	Tiimijohtaminen
2. Suullinen viestintä	Kommunikaatio
3. Projektin laajuuden hallinta	Projektin johtaminen
4. Kuuntelutaito	Kommunikaatio
5. Projektin suunnittelu	Projektin johtaminen
6. Kirjallinen viestintä	Kommunikaatio
7. Vuorovaikutustaidot	Ihmistaidot
8. Kyky motivoida tiimin jäseniä	Tiimijohtaminen
9. Neuvottelutaidot	Ihmistaidot
10. Organisoitukyky	Organisointi
11. Ajankäytön hallinta	Projektin johtaminen
12. Suhteiden rakentaminen	Ihmistaidot
13. Resurssien hallinta	Projektin johtaminen
14. Konfliktien hallinta	Ihmistaidot
15. Riskien hallinta	Projektin johtaminen
16. Kyky huomioida yksityiskohtia	Henkilökohtaiset ominaisuudet
17. Kustannustenhallinta	Projektin johtaminen
18. Kyky tehdä asioita samanaikaisesti, multitasking	Organisointi
19. Analyttiset taidot	Ongelmanratkaisu

Johtamisen taitoa tarvitaan erityisesti projektiin liittyvien henkilöiden motivoimiseen ja ohjaamiseen projektissa, kun taas suullinen viestintä auttaa erityisesti kommunikoimaan sidosryhmien kanssa, minkä lisäksi siitä on apua ongelmien selvittämisessä ja ratkaisemisessa. Projektin laajuuden hallinta on merkittävässä roolissa, sillä tällä saralla tehtävät päätökset vaikuttavat suoraan projektin aikatauluun ja kustannuksiin. Tähän vaikuttaa kyky osata havainnoida ja tarkkailla sidosryhmien odotuksia sekä projektista toimitettavaa tuotetta. (Keil ym. 2013).

Kuunteleminen on kriittistä erityisesti, jotta ollaan perillä sidosryhmistä ja niiden odotuksista sekä huomataan ajoissa projektiin liittyvät ongelmat ja riskit. Projektin suunnitteluun liittyvä osaaminen taas on tärkeässä asemassa sen vuoksi, että huolellinen suunnittelu asettaa raamit projektin budjetille sekä lopputulokselle. Selkeä suunnitelma auttaa hahmottamaan projektin statusta ja tavoitteita, mutta myös tunnistamaan projektin jäsenten vastuualueita. (Keil ym. 2013).

## 5 PROJEKTIPÄÄLLIKÖN TAIDOT OSANA HAJAUTETTUJEN KETTERIEN MENETELMIEN HAASTEISIIN VASTAAMISTA

Kuten aiemmin todettiin, monet hajautetun ketterän ohjelmistokehityksen haasteista liittyvät enemmän etäisyyden aiheuttamiin haasteisiin kuin itse ketteriin toimintatapoihin. Näin ollen monet hajautettujen ketterien ohjelmistoprojektien haasteet ovat samankaltaisia hajautettujen ohjelmistoprojektien sekä globaalien ohjelmistoprojektien haasteiden kanssa.

Keil ym. (2013) mukaan projektipäällikön taidot ovat kriittisiä IT-projektin onnistumisen kannalta, ja erityisesti tietyt IT-alan projektipäällikön taidot korreloivat vahvasti projektin lopputuloksen kanssa. Tämän vuoksi on erityisen tärkeää ymmärtää, mitä taitoja projektipäälliköltä vaaditaan. Tässä luvussa peilataan Keilin ym. (2013) listaa projektipäällikölle tarpeellisista taidoista erityisesti hajautettujen ketterien ohjelmistoprojektien haasteisiin ja pyritään tämän pohjalta nimeämään projektipäällikön taitoja ja ominaisuuksia, joilla on erityisesti merkitystä hajautetun ketterän ohjelmistoprojektin lopputuloksen onnistumisen kannalta.

Alaluvussa 5.1 kerrotaan lyhyesti projektipäällikön roolista ketterissä projekteissa ja alaluvussa 5.2 tarkastellaan hajautetuissa ketterissä projekteissa tarvittavia projektipäällikön taitoja ja osaamista.

### 5.1 Projektipäällikön rooli ketterissä projekteissa

Projektipäällikkö määritellään PMBOKin (The Project Management Book of Knowledge) mukaan henkilöksi, jonka organisaatio on nimennyt johtamaan tiimiä, ja joka on vastuussa projektin tavoitteiden saavuttamisesta. Projektipäällikön roolia ketterissä menetelmissä on kuitenkin kyseenalaistettu, sillä ketterissä menetelmissä korostetaan itseohjautuvia tiimejä sekä nopeaa reagointia muutoksiin. Ketterissä menetelmissä ei myöskään erikseen mainita projektipäällikön roolia, vaan sen sijaan usein käytettäviä termejä ovat scrummaster ja



tuoteomistaja (Shastri, Hoda & Amor, 2016). Tästä huolimatta myös ketteristä projekteista löytyy projektipäälliköksi nimettyjä henkilöitä, ja esimerkiksi Shastrin ym. (2016) tutkimustulosten mukaan 67 % projekteista sisälsi projektipäälliköksi nimetyn henkilön. Saman tutkimuksen mukaan projektipäällikkö löytyy todennäköisemmin niistä projekteista, joissa projektitiimi on suuri, yli 5 henkilöä. Tiimin koon ylittäessä 25 henkilöä oli Shastrin ym. (2016) mukaan projektipäällikön läsnäolo pikemminkin sääntö kuin poikkeus. Näin suuret tiimit kuitenkin jakautuvat Shastrin ym. (2016) mukaan tyypillisesti useisiin pienempiin tiimeihin, mikä osalta lisää tarvetta johtamiselle ja koordinoinnille.

Projektipäällikkö löytyy tyypillisesti myös hajautetuista ketteristä projekteista. Shastrin ym. (2016) tutkimustulosten mukaan projektipäällikkö löytyy todennäköisemmin hajautetusta ketterästä tiimistä kuin paikallisesti toimivasta projektitiimistä. Tämä osaltaan kertoo siitä, että hajautettujen tiimien hallintaan tarvitaan erillinen rooli, joka tyypillisesti nähdään olevan projektipäällikön vastuulla.

Shastri, Hoda ja Amor (2021) ovat uusimmassa tutkimuksessaan selvittäneet tarkemmin projektipäällikön roolia ketterissä projekteissa. Usein ketterissä projekteissa on scrummasterin rooli, jonka tehtävistä osa on Shastrin ym. (2021) mukaan päällekkäisiä tyypillisten projektipäällikön vastuiden kanssa. Tällaisia tehtäviä ovat muun muassa tiimin toiminnan fasilitointi, esteiden poistaminen, tiimin kannustaminen itseohjautuvuuteen, ketterien toimintatapojen noudattamisen varmistaminen sekä tiimin suojaaminen ulkoisilta häiriöiltä. Toisaalta tutkimuksessa löydettiin myös projektipäällikölle ominaisia tehtäviä, jotka eivät kuulu scrummasterille. Näitä olivat muun muassa projektin edistymisen seuranta, projektin tilan raportointi, henkilöiden johtaminen sekä budjetointi. Näitä tehtäviä varten Shastri ym. (2021) toteavat projektipäällikön roolin olevan tarpeellinen. Toisaalta tutkimuksessa tuodaan esiin myös mahdollisuus sille, että projektipäällikkö ja scrummaster voivat olla sama henkilö. Aina näin ei kuitenkaan ole, vaan esimerkiksi tilanteessa, jossa työ ostetaan organisaation ulkopuolelta, voi projektipäällikkö kuulua ostavaan organisaatioon, kun taas scrummaster ja tiimi voivat tulla alihankkijalta.

## **5.2 Projektipäällikön taitojen merkitys hajautettujen projektien haasteiden ratkaisussa**

Tässä luvussa pyritään nimeämään projektipäällikölle tarpeellisia taitoja Rameshin ym. (2006) tunnistamien hajautettujen ketterien menetelmien haasteiden sekä niitä korjaavien käytänteiden pohjalta (taulukko 4). Rameshin ym. (2006) mukaan tietyille haasteille (esimerkiksi kommunikaation tarve vs. kommunikaation haasteet) sopii korjaaviksi toimenpiteiksi useampia erilaisia käytänteitä. Toisaalta joillekin Rameshin ym. (2006) nimeämille haasteille, kuten muodollisen ja epämuodollisen sopimisen ristiriidalle sekä tiimin yhteenkuuluvuuden tunteen puutteelle, soveltuvat korjaukseksi samat käytänteet. Tämä vuoksi pro-

jektipäällikön kompetensseja tarkastellaan tässä osiossa pääsääntöisesti Rameshin ym. (2006) nimeämien korjaavien käytänteiden pohjalta yksittäisten haasteiden sijaan toiston välttämiseksi.

TAULUKKO 4 Rameshin ym. (2006) hajautettujen ketterien projektien haasteet sekä niihin vaikuttavat projektipäällikön taidot

Haaste (Ramesh ym., 2006)	Korjaavat käytänteet (Ramesh ym., 2006)	Keskeiset projektipäällikön taidot (Keil ym., 2013)
Kommunikaation tarve vs. kommunikaation haasteet	Viestintää voidaan kehittää työaikoja yhtenäistämällä, epämuodollista viestintää lisäämällä, riittävällä koordinoinnilla sekä jatkuvalla kommunikoinnilla. Kommunikointiin voidaan vaikuttaa myös tietämyksen jakamista edistämällä, kuten ylläpitämällä tietopankkia, keskittymällä projektin alussa selkeimpien toiminnallisuuksien toteuttamiseen haastavimpien sijaan sekä pitämällä kehityssykliä lyhyinä, mutta ilman aikaikkunaa.	Suullinen viestintä Kirjallinen viestintä Kuuntelutaito Etätiimien hallinta Kyky rakentaa yhteyksiä eri tiimien välille Kyky motivoida tiimin jäseniä Kyky ymmärtää liiketoiminta-alueita Organisointikyky
Ennalta määritetyt vs. muuttuvat laatuvaatimukset	”Luota mutta varmista”-periaatteen noudattaminen. Hajautetaan laadunvarmistus esimerkiksi paikallisen tiimin ja etätiimien kesken. Täydennetään epämuodollista viestintää tarvittavalla määrällä dokumentaatiota.	Laatukeskeisyys
Ihmiskeskeinen vs. prosessikeskeinen ohjaus	Myös ohjauksen haasteiden apuna voidaan käyttää ”luota mutta varmista”-periaatetta (ks. yllä). Lisäksi prosessin jatkuva mukauttaminen, kuten vaatimusten ja suunnitelmien muutosten huomiointi osana iteraatioita sekä vaatimusten dokumentointi eri muodollisuuden tasoilla.	Projektin suunnittelu Projektin laajuuden hallinta
Muodollinen vs. epämuodollinen sopiminen	Luottamuksen rakentaminen lisäämällä sidosryhmien ja rahoittajien vierailuja sekä tiimin yhtenäisen kulttuurin rakentaminen.	Yhteistyö Saavutusten huomiointi Organisointikyky
Tiimin yhteenkuuluvuuden tunteen puute	Myös yhteenkuuluvuuden tunnetta voidaan edistää luottamusta rakentamalla (ks. yllä).	Yhteistyö Saavutusten huomiointi Organisointikyky

Kommunikaation haasteiden lieventämiseksi Ramesh ym. (2006) nostavat esiin tarpeen viestinnän kehittämiseksi. Muun muassa työaikojen yhtenäistäminen nostetaan Rameshin ym. (2006) tutkimuksessa eräksi käytänteeksi, joka helpot-

taa tiimien viestintää. Yhtenäiset työajat mahdollistavat nopeamman viestinnän, kun vastauksissa ei ole viivettä, ja toisaalta myös palaverien järjestäminen helpottuu. Yhtenäisten työaikojen järjestäminen vaatii kuitenkin projektipäälliköltä etätiimien hallinnan taitoa, kuten kykyä ottaa huomioon kansalliset vapaapäivät, loma-ajat sekä aikavyöhykkeistä johtuvat haasteet.

Etätiimien hallinta onkin Keilin ym. (2013) listauksessa on mainittu eräänä projektipäällikön taitona, joka on aiempiin vastaaviin tutkimuksiin verrattuna täysin uusi projektipäällikön kannalta huomioon otettava taito (Keil ym., 2013). Kasvava hajautettujen ohjelmistoprojektien määrä on todennäköisesti vaikuttanut siihen, että etätiimien hallinta on noussut Keilin ym. (2013) tutkimuksessa tiimin johtamisen taitokategorian alle. Keilin ym. (2013) mukaan projektipäällikön on erityisen tärkeää osata ottaa hajautetuissa projekteissa huomioon eri aikavyöhykkeillä työskentelevät tiimin jäsenet sekä erilaiset kulttuurit ja niihin liittyvät tavat, perinteet, lomapäivät ja työetiikka.

Väärinkäsitykset ovat tyypillinen ongelma, joka korostuu erityisesti hajautetussa ympäristössä työskenneltäessä, kun mahdollisuuksia epämuodolliselle viestinnälle on vähemmän. Väärinkäsitysten vähentämiseksi hajautetuissa tiimeissä voidaan päätyä ratkaisuun, jossa esimerkiksi projektin vetäjä tai projektipäällikkö toimii eri toimipisteiden välisen viestinnän keskushenkilönä ja fasilitoi tiimien välistä kommunikaatiota (Ramesh ym., 2006). Tämä korostaa entisestään projektipäällikön kommunikointitaitoja, kuten suullista ja kirjallista viestintää sekä kuuntelutaitoa.

Kun tyypillisesti ketterässä kehityksessä projektipäällikön suorittama tiimin ja sen tehtävien koordinointi pidetään minimissään, hajautetussa ympäristössä koordinoinnille on enemmän tarvetta (Ramesh ym., 2006). Koordinointi edellyttää projektipäälliköltä enemmän organisointikykyä kuin paikallisesti toteutettavat ketterät ohjelmistoprojektit. Samalla se auttaa tiimiä ja projektipäällikköä pysymään paremmin perillä siitä, mitä tehtäviä kenenkin vastuulla on ja miten ne etenevät suhteessa projektin tavoitteisiin. Projektipäällikön on lisäksi tärkeää osata rakentaa yhteyksiä eri tiimien välille eli huomioida, mitkä ovat minkäkin tiimin vastuut, missä on päällekkäisyyksiä ja onko vastuiden jaossa mahdollisesti jäänyt aukkoja (Keil ym., 2013).

Myös tietämyksen jakamisen edistäminen nähdään Rameshin ym. (2006) tutkimuksessa kommunikaation haasteita korjaavana käytänteenä. On tärkeää, että tiimien jäsenillä on pääsy työn kannalta kriittiseen tietoon. Tätä varten Ramesh ym. (2006) ehdottavat muun muassa tuotetta ja prosesseja koskevan tietopankin ylläpitämistä. Tietämyksen jakaminen vaatii projektipäälliköltä muun muassa tiimijohtamisen taitoja. Projektin alussa projektipäällikkö voi esimerkiksi pyrkiä motivoimaan tiimiä siten, että ensin keskitytään toteuttamaan selkeitä toiminnallisuuksia ja sitä kautta keräämään osaamista sekä ymmärrystä projektin aihealueesta, vaikka tyypillisesti ketterissä menetelmissä suositellaankin lähtemään liikkeelle kriittisten ominaisuuksien toteutuksesta.

Ramesh ym., (2006) tuovat tutkimuksessaan esiin tarpeen lyhyille kehitysyksiköille, joille ei ole määritelty aikaikkunaa. Tämä käytäntö perustellaan tutkimuksessa sillä, että hajautetuissa ketterissä projekteissa on usein alussa haas-

teita hahmottaa riittävällä tasolla liiketoiminta-aluetta ja asiakkaan tarpeita. Myös asiakkaan nimeämät vaatimukset voivat vielä projektin alussa olla liian laaja-alaisia, jotta niiden pohjalta olisi lyhyen iteraation aikana mahdollista saada toteutettua kokonaista ominaisuutta. Tämän vuoksi projektin alussa voi olla tarpeen toteuttaa lyhyitä syklejä ilman aikaikkunaa, jotta kehittäjät saavat riittävän kuvan toiminnallisuuksista sekä liiketoiminta-alueesta. Jotta tiimi saa oikean käsityksen liiketoiminta-alueesta, tulee myös projektipäällikön tuntee liiketoiminta-alue hyvin projektin alusta lähtien, jotta hän pystyy johtamaan ja ohjaamaan tiimiä oikeaan suuntaan.

Projektipäällikön tulee huomioida myös laadun merkitys hajautetuissa projekteissa. Projektipäällikön vastuulla on esimerkiksi asettaa laatustandardit ja varmistaa, että tiimi noudattaa niitä (Keil ym., 2013). Projektipäällikön laatu-keskeisyys onkin tärkeä taito, joka ilmenee hajautetuissa ketterissä projekteissa erityisesti osana Rameshin ym. (2006) esittelemää ”Luota mutta varmista”-periaatetta, johon sisältyy kaksi käytännettä, hajautettu laadunvarmistus sekä epämuodollisen viestinnän täydentäminen dokumentaatiolla. Nämä käytänteet auttavat lisäämään tiimien välistä luottamusta, mutta samalla ne antavat mahdollisuuden tarkastella tuotteen ja prosessien laatua. Ketterien menetelmien periaatteista poiketen tällaiset muodolliset tavat varmistaa tuotteiden ja prosessien laatua on Rameshin ym. (2006) tutkimuksessa koettu tarpeellisiksi hajautetussa ympäristössä työskenneltäessä. Hajautetun laadunvarmistuksen ideana on, että esimerkiksi paikallinen laadunvarmistustiimi tarkastaa ulkoistetun tiimin käytänteiden sekä toteutusten laadun. Epämuodollisen viestinnän täydentäminen dokumentaatiolla sen sijaan on tarpeen erityisesti sen vuoksi, että erityisesti monimutkaisimmista ja kriittisimmistä artefakteista on olemassa myös riittävä määrä dokumentaatiota epävirallisen tiedon lisäksi. ”Luota mutta varmista” -periaate auttaa Rameshin ym. (2006) mukaan erityisesti ennalta määritettyjen ja muuttuvien laatuvaatimusten sekä ihmiskeskeisen ja prosessikeskeisen ohjauksen aiheuttamiin ristiriitoihin.

Ramesh ym. (2006) nostavat tutkimuksessaan prosessin jatkuvan mukauttamisen ratkaisuksi ihmiskeskeisen ja prosessikeskeisen ohjauksen aiheuttamaan haasteeseen. Prosessin mukauttaminen vaatii jokaisen iteraation aikana vaatimusten sekä suunnitelmien huomioon ottamista. Projektipäällikön kannalta prosessien mukauttaminen vaatii erityisesti osaamista projektin suunnittelun sekä projektin laajuuden hallinnan osalta. On tärkeää osata tiedostaa, missä vaiheessa projektia kulloinkin ollaan tavoitteeseen nähden.

Ramesh ym. (2006) nimeävät tiimin yhtenäisen kulttuurin rakentamisen tärkeäksi osaksi luottamuksen rakentamista. Luottamuksen rakentaminen taas on tutkimuksessa nimetty korjaavaksi käytänteeksi sekä tiimin yhteenkuuluvuuden tunteen puuttumiseen että muodollisen ja epämuodollisen sopimisen aiheuttamaan ristiriitaan. Myös Verburg, Bosch-Sijtsema ja Vartiainen (2013) ovat virtuaalisessa ympäristössä tehtävän projektityön kriittisiä menestystekijöitä tutkiessaan todenneet tiimin jäsenten välillä vallitsevalla luottamuksella ja kommunikaatiolla olevan erittäin suuri merkitys hajautetun projektin onnistumiselle.

Tiimin yhtenäisen kulttuurin rakentamisessa projektipäälliköllä on merkittävä rooli aina tiimin jäsenten valinnasta alkaen. Rameshin ym. (2006) tutkimuksessa nousi esiin pyrkimys välttää tiimin yhteenkuuluvuuteen liittyviä haasteita valitsemalla lähtökohtaisesti tiimeihin sellaisia henkilöitä, jotka ovat jo aiemmin työskennelleet yhdessä, ja joiden yhteinen osaaminen on projektin toteutuksen kannalta tarpeeksi laaja. Tämä edellyttää projektipäälliköltä organisointikykyä. Tiimin muodostamisen lisäksi yhteishengen rakentamisessa on tärkeää, että projektipäällikkö osaa huomioida ja nostaa esiin tiimin saavutuksia, sillä se osaltaan motivoi ja kannustaa koko tiimiä. Myös projektipäällikön halukkuus ja kyky tehdä yhteistyötä tiimin kanssa ja olla mukana ratkaisujen kehittämässä edistää yhtenäisen kulttuurin rakentumista ja toiminnan tehokkuutta sekä tuottavuutta (Keil ym., 2013).

Rameshin ym. (2006) lisäksi muun muassa Langer, Slaughter ja Mukhopadhyay (2014) ovat tutkimuksessaan huomioineet hajautetun ohjelmistokehityksen mukanaan tuomia haasteita. Poiketen Rameshin ym. (2006) tuloksista Langerin ym. (2014) tutkimuksessa nostetaan esiin erityisesti se, että hajautetuissa ohjelmistoprojekteissa voi ilmetä kriittisiä ja yllättäviä ongelmia tavallista enemmän johtuen tiedonkulun rajoittumisesta. Tutkimuksen mukaan erityisesti projektipäälliköllä on tärkeä rooli tällaisten yllättävien ja nopeaa reagointia vaativien ongelmien ratkaisussa. Langer ym. (2014) huomioivat tällaisten tilanteiden varalle tärkeäksi projektipäällikön ominaisuudeksi käytännön älykkyyden (practical intelligence), mitä ei kuitenkaan suoraan tässä muodossa esiinny Keilin ym. (2013) projektipäällikön taitojen listauksessa. Sen sijaan Keilin ym. (2013) taitolistauksessa yhtenä kategoriana on ongelmanratkaisutaidot, joihin sisältyy kyky tunnistaa, analysoida ja ratkaista ongelmia.

Myös Cascio ja Shurygailo (2003) ovat tarkastelleet hajautettuja tiimejä erityisesti siltä kannalta, mitä haasteita maantieteellinen ja ajallinen ero tiimin jäsenten välillä aiheuttaa tiimin johtamiselle. Sekä fyysisesti samassa paikassa työskenteleville tiimeille että etänä työskenteleville tiimeille projektipäällikön toiminnalla on suuri merkitys, mutta Cascion ja Shurygailon (2003) mukaan mikään tietty johtamistyyli ei välttämättä ole paras virtuaaliselle tiimille. Jokaisen johtajan tulee tarkastella hajautettuna tapahtuvan työn järjestelyjen vaikutusta omaan johtamistaansa, ja sopeutua tilanteeseen sen mukaisesti.

## 6 YHTEENVETO

Tässä tutkimuksessa on käsitelty hajautetun ketterän ohjelmistokehityksen aiheuttamia haasteita sekä sitä, mitkä taidot ovat erityisen tärkeitä projektipäällikölle hajautettuja ketteriä projekteja johtaessa. Kirjallisuuden perusteella monet hajautettuun ketterään ohjelmistokehitykseen liittyvistä haasteista koskevat erityisesti etäisyyden aiheuttamia ongelmia ja kommunikaatiota, mutta myös ketterät menetelmät sekä niiden peruseräpäätet tuovat omat haasteensa hajautettujen ketterien ohjelmistoprojektien toteutukseen.

Ketterissä menetelmissä korostetaan kasvokkain tapahtuvan kommunikoinnin merkitystä, mikä on hajautetussa ympäristössä haastavaa toteuttaa johduen aikaeroista sekä maantieteellisestä etäisyydestä. Ketterät menetelmät nostavat tärkeäksi ominaisuudeksi myös tiimien itseohjautuvuuden, mikä taas on pienemmässä roolissa hajautetussa ympäristössä, jossa tyypillisesti tarvitaan enemmän koordinoimista eri sijainneissa toimivien tiimien sekä tiiminjäsenten välille. Myös dokumentaatiolle on enemmän tarvetta hajautetuissa projekteissa rajattujen kommunikointimahdollisuuksien seurauksena.

Toisaalta kirjallisuuden perusteella ketteristä menetelmistä, kuten Scrumista, voi olla hajautetuissa ohjelmistoprojekteissa jopa etua (Alzoubi ym., 2014; Paasivaara & Lassenius, 2006; Paasivaara ym., 2008; Paasivaara ym., 2009). Ketterät menetelmät korostavat kommunikaation merkitystä, ja esimerkiksi Scrumin päivittäispalaverit sekä muut tapahtumat pakottavat kehitystiimin kommunikoimaan tiiviisti keskenään. Tämä voi edistää hajautettujen tiimien kommunikaatiota, kun "pakotettu" kommunikointi madaltaa kynnystä myös muulle viestinnälle. Scrumista on myös kehitetty erilaisia versioita hajautettuja projekteja varten, kuten Scrum of Scrums sekä integroidut Scrum-tiimit (Sutherland ym., 2007).

Tässä tutkielmassa on esitetty kaksi tutkimuskysymystä, joista toinen on "Mitä haasteita hajautetussa ketterässä ohjelmistokehityksessä on ja mitä niiden korjaamiseksi voidaan tehdä?". Tähän on tutkielmassa vastattu Rameshin ym. (2006) hajautetun ketterän ohjelmistokehityksen haasteiden jaottelelu mukailleen. Koska tutkielman tarkoituksena on keskittyä erityisesti hajautetun ketterän ohjelmistokehityksen haasteisiin, on haasteiden erittelyssä pääasi-

allisena lähteenä käytetty kyseistä Rameshin ym. (2006) luomaa erittelyä hajautettujen ketterien ohjelmistoprojektien haasteista. Monissa muissa tutkimuksissa hajautetun ketterän ohjelmistokehityksen haasteet ja hajautetun ohjelmistokehityksen sekä globaalin ohjelmistokehityksen haasteet on yhdistetty yhdeksi kokonaisuudeksi, jolloin kaikki haasteet eivät ole suoraan seurausta ketterien periaatteiden yhdistämisestä hajautettuihin projekteihin. Esimerkiksi Kajko-Mattssonin ym. (2010) tutkimuksessa nostettiin hajautettujen ketterien ohjelmistoprojektien yhteydessä esiin myös sellaisia haasteita, joita esiintyy muutoinkin kuin hajautetuissa projekteissa.

Hajautetun ketterän ohjelmistokehityksen haasteet jakautuvat Rameshin ym. (2006) jaottelun mukaisesti kolmeen osa-alueeseen: kommunikaation, kontrollin sekä luottamuksen aiheuttamiin haasteisiin. Kontrolliin liittyvät haasteet jakautuvat edelleen ennalta määrättyjen ja muuttuvien laatuvaatimusten sekä ihmiskeskeisen ja prosessikeskeisen ohjauksen aiheuttamiin ristiriitoihin, kun taas luottamuksen aiheuttamiin haasteisiin sisältyy tiimin yhteenkuuluvuuden tunteen puute sekä muodollisen ja epämuodollisen sopimisen aiheuttama ristiriita. Näiden haasteiden korjaamiseksi hajautetuissa ketterissä projekteissa tulee Rameshin ym. (2006) mukaan kiinnittää huomiota viestinnän kehittämiseen, tietämyksen jakamisen edistämiseen, prosessin jatkuvaan mukauttamiseen sekä luottamuksen rakentamiseen.

Hajautettujen ketterien ohjelmistoprojektien haasteiden ja niitä korjaavien käytänteiden sekä Keilin ym. (2013) esittelemän projektipäällikön keskeisten taitojen listauksen pohjalta tässä tutkielmassa on selvitetty toisen tutkimuskysymyksen mukaisesti, mitkä taidot ja osaamisalueet ovat IT-alan projektipäällikölle erityisen tärkeitä hajautetuissa ketterissä ohjelmistoprojekteissa. Viestintään liittyvien haasteiden ratkaisemisessa korostuvat projektipäällikön kommunikointitaidot, kuten suullinen ja kirjallinen viestintä sekä kuuntelutaito. Myös etätiimien hallinta nousee tärkeäksi taidoksi erityisesti, mikäli projekti on hajautunut maantieteellisesti laajalle alueelle. Tällöin on tärkeää tiedostaa aikeroihin sekä kulttuuriin ja toimintatapoihin liittyvät eroavaisuudet. Kommunoinnin haasteet aiheuttavat tarvetta myös koordinoinnille, jolloin projektipäällikön organisointitaito sekä kyky luoda yhteyksiä tiimien välille korostuu. Tietämyksen jakamisen edistämässä projektipäällikölle on erityisesti hyötyä siitä, että hän kykenee motivoimaan tiimin jäseniä sekä ymmärtää liiketoimintaa. Projektipäällikön tulee olla myös laatuksellinen, mitä korostaa erityisesti Rameshin ym. (2006) ehdottama hajautetun laadunvarmistuksen käytänne. Myös projektipäällikön osaaminen projektin suunnittelun sekä projektin laajuuden hallinnan saralta on merkittävässä osassa erityisesti, koska prosessin mukauttamisen seurauksena iteraatioissa tulee jatkuvasti ottaa huomioon muuttuvat vaatimukset ja suunnitelmat. Luottamuksen aiheuttamien haasteiden suhteen projektipäällikön taidoista korostuu yhteistyö, saavutusten huomiointi sekä organisointikyky.

IT-alan projektipäällikön taitojen tärkeyttä on vertailtu paljon kirjallisuudessa, mutta erityisesti hajautettujen ketterien ohjelmistoprojektien projektipäälliköiden taitoja tai kompetensseja koskevaa tutkimusta ei aineistohaassa

löytynyt. Tämä tutkimus avaakin näkökulmaa siihen, miten projektipäällikön taidot ja osaaminen suhteutuvat hajautettujen ketterien projektien läpivientiin.

Tyypillinen piirre jo olemassa olevalle tutkimukselle IT-alan projektipäällikön taidoista on se, että muun muassa ihmistaitoja, kommunikointia sekä projektin, ihmisten ja tiimien johtamista pidetään yleisesti huomattavasti tärkeämpinä kuin esimerkiksi teknisiä taitoja (El-Sabaa, 2001; Keil ym., 2013; Medina & Francis, 2015; Silva de Araújo & Pedron, 2015; Stevenson & Starkweather, 2010). Tämän tutkimuksen pohjalta vaikuttaa siltä, että ”pehmeät taidot” korostuvat myös hajautetuissa ketterissä projekteissa, joskin ihmistaidoista mikään ei erityisesti painottunut tämän tutkimuksen yhteydessä.

Valtaosa tässä tutkimuksessa esiin nousseista taidoista sisältyy Keilin ym. (2013) jaottelun mukaisesti kommunikaatioon (suullinen viestintä, kirjallinen viestintä, kuuntelutaito), tiimijohtamiseen (etätiimien hallinta, kyky rakentaa yhteyksiä eri tiimien välille, kyky motivoida tiimin jäseniä, yhteistyö, saavutusten huomiointi), organisointiin (organisointikyky) ja projektinhallintaan (projektin suunnittelu, projektin laajuuden hallinta), mutta yksittäisinä osa-alueina esiin nousivat myös liiketoiminta-alueen tuntemus (kyky ymmärtää liiketoiminta-aluetta) sekä ammatillisuus (laatu keskeisyys). Voikin olla, että hajautetuissa ketterissä ohjelmistoprojekteissa korostuvat jossain määrin samat ongelmat, joita kohdataan myös muissa ohjelmistoprojekteissa, mutta etäisyys tuo tällaisia haasteita vahvemmin esiin.

Erityisesti tässä tutkimuksessa korostuu projektipäällikön taitojen osalta tiimijohtamisen merkitys, jonka alta löytyi Rameshin ym. (2006) ja Keilin ym. (2013) tutkimuksia yhdistämällä viisi tärkeää projektipäällikön taitoa. Myös kommunikoinnin alta löytyi kolme taitoa ja projektinhallinnan alta kaksi, minkä lisäksi organisointikyky nousi keskeiseksi taidoksi useiden haasteiden yhteydessä. Kuten Keil ym. (2013) kuvaavat tiimijohtamisen taitoja, tarvitaan niitä erityisesti projektin jäsenten motivointiin sekä johtamiseen niin tiimien välillä kuin tiimien sisällä. Koska hajautetuissa projekteissa tiimit ja niiden jäsenet toimivat eri sijainneista käsin, ei olekaan kovin yllättävää, että tiimijohtamisen ja kommunikoinnin osa-alueet korostuvat tutkimustuloksissa.

Tutkimus pohjautuu pitkälti Rameshin ym. (2006) esille nostamiin hajautettujen ketterien projektien haasteisiin sekä Keilin ym. (2013) tutkimuksessa esiteltyihin IT-alan projektipäällikön taitoihin. Nämä tutkimukset on valittu pääasiallisiksi lähteiksi sen perusteella, että Keilin ym. (2013) tutkimuksessa on kattava lista projektipäällikön taidoista ja Rameshin ym. (2006) tutkimus rajaa projektien haasteet täsmällisimmin koskemaan ainoastaan ketteryuden ja hajauttamisen yhdistelmää. Näin tiukka lähdeaineiston rajaaminen voi kuitenkin osaltaan rajoittaa tutkimustulosten yleistettävyyttä, ja eri projekteissa voi nousta esiin myös muita haasteita, joihin on mahdollisesti myös erilaisia ratkaisuja. Jatkotutkimuksessa voisi olla hyvä ottaa huomioon laajempi katsaus hajautettujen ketterien projektien haasteita.

Tässä tutkimuksessa projektipäällikölle tarpeellisia taitoja on nostettu esiin ainoastaan kirjallisuuskatsauksen pohjalta, eikä taitoja ole tämän vuoksi voitu laittaa tärkeysjärjestykseen. Jatkotutkimuksena voitaisiinkin selvittää esimer-



kiksi haastatteluiden tai kyselytutkimuksen avulla, miten merkittäviksi tässä tutkimuksessa esiin nostetut IT-alan projektipäällikön taidot todellisuudessa koetaan hajautettujen ketterien ohjelmistoprojektien kontekstissa ja nouseeko näiden lisäksi esiin muita projektipäällikön taitoja, jotka koetaan erityisen tärkeiksi verrattuna muihin projekteihin. Mielenkiintoinen tutkimusaihe voisi olla myös, millainen merkitys projektitiimin koolla on hajautettuun ketterään ohjelmistokehitykseen ja projektin johtamiseen.

Tämän tutkimuksen tuloksia voidaan hyödyntää esimerkiksi työelämässä, kun on tarve toteuttaa ketteriä projekteja hajautetusti. Tutkimustulokset antavat viitteitä siitä, millaisia ominaisuuksia ja osaamista tällaisen projektin projektipäälliköllä olisi erityisesti syytä olla, ja mitä taitoja mahdollisesti kannattaa kehittää. Tutkimuksen tulokset auttavat myös hahmottamaan, mihin haasteisiin hajautetuissa ketterissä projekteissa kannattaa erityisesti kiinnittää huomiota.

## LÄHTEET

- Agile Alliance. (2001). Manifesto for Agile Software Development. Haettu 3.12.2020 osoitteesta <https://agilemanifesto.org>.
- Alzoubi, Y. I., Gill, A. Q. & Al-Ani, A. (2016). Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information & Management*, 53(1), 22-37.
- Beecham, S., Noll, J. & Richardson, I. (2014). Using Agile practices to solve Global Software Development problems - A Case Study. Teoksessa 2014 *IEEE International Conference on Global Software Engineering Workshops* (5-10). IEEE.
- Berczuk, S. (2007). Back to Basics: The Role of Agile Principles in Success with an Distributed Scrum Team. Teoksessa *Agile 2007 (AGILE 2007)* (382-388). IEEE.
- Carmel, E. & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2), 22-29.
- Cascio, W. F. & Shurygailo, S. (2003). E-Leadership and virtual teams. *Organizational Dynamics*, 31(4), 362-376.
- Dillon, S. & Taylor, H. (2015). Employing Grounded Theory to Uncover Behavioral Competencies of Information Technology Project Managers. *Project Management Journal*, 46(4), 90-104.
- Ebert, C. & Paasivaara, M. (2017). Scaling Agile. *IEEE Software*, 34(6), 98-103.
- El-Sabaa, S. (2001). The skills and career path of an effective project manager. *International journal of project management*, 19(1), 1-7.
- Holmström, H., Fitzgerald, B., Ågerfalk, P. J. & Conchúir, E. Ó. (2006). Agile Practices Reduce Distance in Global Software Development. *Information Systems Management*, 23(3), 7-18.
- Jalali, S. & Wohlin, C. (2010). Agile Practices in Global Software Engineering - A Systematic Map. Teoksessa 2010 *5th IEEE International Conference on Global Software Engineering* (45-54). IEEE.
- Kajko-Mattsson, M., Azizyan, G. & Magarian, M. K. (2010). Classes of Distributed Agile Development Problems. Teoksessa 2010 *Agile Conference* (51-58). IEEE.

- Karaman, K. & Kurt, M. (2015). Comparison of project management methodologies: prince 2 versus PMBOK for it projects. *Int. Journal of Applied Sciences and Engineering Research*, 4(4), 572-579.
- Keil, M., Lee, H. K. & Deng, T. (2013). Understanding the most critical skills for managing IT projects: A Delphi study of IT project managers. *Information & Management*, 50(7), 398-414.
- Langer, N., Slaughter, S. A. & Mukhopadhyay, T. (2014). Project Managers' Practical Intelligence and Project Performance in Software Offshore Outsourcing: A Field Study. *Information Systems Research*, 25(2), 364-384.
- Lee, G. & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114.
- Lee, S., Yong, H.-S. (2010). Distributed agile: project management in a global environment. *Empirical Software Engineering*, 15, 204-217.
- Lous, P., Tell, P., Michelsen, C. B., Dittrich, Y. & Ebdrup, A. (2018). From Scrum to Agile: a Journey to Tackle the Challenges of Distributed Development in an Agile Team. Teoksessa *ICSSP '18: Proceedings of the 2018 International Conference on Software and System Process* (s. 11-20). Association for Computing Machinery.
- Matalonga, S., Solari, M. & Matturro, G. (2013). Factors affecting distributed Agile projects: A systematic review. *International Journal of Software Engineering and Knowledge Engineering*, 23(9), 1289-1301.
- Medina, A. & Francis, A. J. (2015). What Are the Characteristics that Software Development Project Team Members Associate with a Good Project Manager?. *Project Management Journal*, 46(5), 81-93.
- Napier, N. P., Keil, M. & Tan, F. B. (2009). IT project managers' construction of successful project management practice: a repertory grid investigation. *Information Systems Journal*, 19(3), 255-282.
- Olson, M. & Olson, S. (2000). Distance matters. *Human-Computer Interaction*, 15(2), 139-178.
- Paasivaara, M., Durasiewicz, S. & Lassenius, C. (2008). Distributed Agile Development: Using Scrum in a Large Project. Teoksessa *2008 IEEE International Conference on Global Software Engineering* (87-95). IEEE.
- Paasivaara, M., Durasiewicz, S. & Lassenius, C. (2009). Using scrum in distributed agile development: A multiple case study. Teoksessa *2009 Fourth IEEE International Conference on Global Software Engineering* (195-204). IEEE.

- Paasivaara, M. & Lassenius, C. (2006). Could Global Software Development Benefit from Agile Methods?. Teoksessa *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)* (109-113). IEEE.
- Petersen, K. & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, 82(9), 1479-1490.
- Phalnikar, R., Deshpande, V. S. & Joshi, S. D. (2009). Applying Agile Principles for Distributed Software Development. Teoksessa *2009 International Conference on Advanced Computer Control* (535-539). IEEE.
- Ramesh, B., Cao, L., Mohan, K. & Xu, P. (2006). Can distributed software development be agile?. *Communications of the ACM*, 49(10), 41-46.
- Saeeda, H., Arif, F., Minhas, N. M. & Humayun, M. (2015). Agile Scalability for Large Scale Projects: Lessons Learned. *Journal of Software*. 10(7), 893-903.
- Schwaber, K. (1997). Scrum development process. Teoksessa *Business Object Design and Implementation* (s. 117-134). Springer London.
- Schwaber, K. & Sutherland, J. (2020). The Scrum Guide. Haettu 10.12.2020 osoitteesta <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>.
- Shastri, Y., Hoda, R. & Amor, R. (2016). Does the "Project Manager" Still Exist in Agile Software Development Projects?. Teoksessa *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)* (57-64). IEEE.
- Shastri, Y., Hoda, R. & Amor, R. (2021). The role of the project manager in agile software development projects. *Journal of Systems and Software, Volume 173*. Julkaistu verkossa 25.11.2020. Haettu 16.12.2020 osoitteesta <https://doi.org/10.1016/j.jss.2020.110871>
- Silva de Araújo, C. C. & Pedron, C. D. (2015). IT project manager competencies and IT project success: a qualitative study. *Organisational Project Management*, 2(1), 53-75.
- Stevenson, D. H. & Starkweather, J. A. (2010). PM critical competency index: IT execs prefer soft skills. *International Journal of Project Management*, 28(7), 663-671.
- Sutherland, D., Viktorov, A., Blount, J. & Puntikov N. (2007). Distributed Scrum: Agile Project Management with Outsourced Development Teams. Teoksessa *2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07)* (274). IEEE.

Verburg, R. M., Bosch-Sijtsema, P. & Vartiainen, M. (2013). Getting it done: Critical success factors for project managers in virtual work settings. *International Journal of Project Management*, 31(1), 68-79.