

# **Tiimien autonomia laajan skaalan ketterässä ohjelmistokehityksessä**

Valtteri Ingervo

**Perustieteiden korkeakoulu**

Kandidaatintyö

Espoo 03.05.2022

**Vastuupettaja**

Stina Giesecke

**Työn ohjaaja**

Casper Lassenius

Copyright © 2022 Valtteri Ingervo

---

**Tekijä** Valtteri Ingervo

---

**Työn nimi** Tiimien autonomia laajan skaalan ketterässä ohjelmistokehityksessä

---

**Koulutusohjelma** Perustieteiden kandidaattiohjelma

---

**Pääaine** Informaatioverkostot

**Pääaineen koodi** SCI3026

---

**Vastuupettaja** Stina Giesecke

---

**Työn ohjaaja** Casper Lassenius

---

**Päivämäärä** 03.05.2022

**Sivumäärä** 35

**Kieli** suomi

---

## Tiivistelmä

Ketterissä ohjelmistokehityksen menetelmissä rakennetaan ohjelmistoa iteratiivisesti autonomisten ja monialaisten ketterien tiimien avulla yhteistyössä ohjelmiston loppukäyttäjien kanssa. Laajan skaalan ketterässä ohjelmistokehityksessä useampi ketterä tiimi työskentelee yhteisen tavoitteen eteen, kun taas autonomisilla tiimeillä viitataan tiimeihin, joilla on vahva auktoriteetti ja vastuu monista työn aspekteista. Kun ketteriä menetelmiä käytetään ohjelmistokehitykseen laajassa skaalassa, tiimien linjausta ja ylhäältä tulevaa koordinoitua tarvitaan keskinäisten riippuvuuksien hallinnoimiseksi ja laadun varmistamiseksi. Tämä on uhka tiimien autonomialle, joka on avaintekijä ketterien menetelmien tehokkuudelle. Toisaalta liika ketterien tiimien autonomia laajassa skaalassa voi johtaa haittoihin kuten yhteisten tavoitteiden hylkäämiseen ja koordinoinnin vaikeutumiseen. Tässä tutkimuksessa selvitettiin yleisimpiä haaste- ja menestystekijöitä tiimien autonomialle laajan skaalan ketterässä ohjelmistokehityksessä, ja pohdittiin niiden pohjalta tasapainoa tiimien autonomian ja yhteisen linjauksen välille. Haaste- ja menestystekijöitä kirjattiin kymmenestä tutkimuksesta muodostaen tekijöistä teemat. Hackmanin työn tekemisen funktiot linkitettiin näihin teemoihin. Haasteteemoja löytyi kuusi, joista yleisin oli *vaikeus rajata tiimien työtä itsenäisiin osiin*. Menes-

tystekijöistä löytyi neljä teemaa, joista yleisimmät olivat *luottamus tiimeihin ja valtuudet tehdä päätöksiä* ja *autonomiia edistävät käytännöt*. Haaste- ja menestystekijöiden teemat linkittyivät vahvasti Hackmanin työn tekemisen funktioon, jossa jonkun pitää hahmottaa, monitoroida ja hallinnoida työprosessia. Tämä osaltaan vahvistaa aiempien tutkimuksien tuloksia, jossa on löydetty keskijohdon pitävän kiinni perinteisestä käskevästä ja kontrolloivasta johtamistyylistä, vaikka organisaatio pyrki ottamaan ketteriä menetelmiä käyttöön. Tasapaino autonomian ja linjauksen välillä voisi löytyä yhteisöllisestä suunnittelusta ja päätöksenteosta. Myös mitä kokeneempi ja taitavampi tiimi on sitä enemmän se voi mahdollisesti hyötyä autonomiasta tarviten vähemmän ylhäältä tulevaa koordinoitua. Lisääntyneen autonomian antaminen tiimeille voi käänteisesti kiihdyttää sen kehittymistä, jos tiimi saa samalla tarpeeksi tukea ja resursseja.

---

**Avainsanat** ketterä kehitys, ohjelmistokehitys, laaja skaala, tiimien autonomia

---

# Sisällys

<b>Tiivistelmä</b>	<b>3</b>
<b>Sisällys</b>	<b>5</b>
<b>1 Johdanto</b>	<b>6</b>
<b>2 Tausta ja termistö</b>	<b>9</b>
2.1 Ketterien menetelmien kehitys vastauksena vesiputousmallille . . . .	9
2.2 Ketterien menetelmien skaalauksen haasteet . . . . .	10
2.3 Autonomian hyödyt ja tarpeet sen rajoitukselle . . . . .	11
2.4 Laajan skaalan määritelmä . . . . .	12
2.5 Organisaatiollisen työn neljä eri funktiota . . . . .	13
<b>3 Aineisto ja menetelmät</b>	<b>14</b>
<b>4 Tulokset</b>	<b>17</b>
4.1 Haasteet . . . . .	17
4.2 Menestystekijät . . . . .	22
<b>5 Pohdinta</b>	<b>26</b>
5.1 Tiimien autonomian tasapaino . . . . .	27
5.2 Tulosten merkitys ja aiheet jatkotutkimukselle . . . . .	28
5.3 Käytännön ohjeita johtajille . . . . .	29
<b>6 Yhteenveto</b>	<b>30</b>

# 1 Johdanto

Ohjelmistot ovat kasvavassa määrin osa arkeamme ja näin kysyntä niiden tehokkaan kehityksen mahdollistamille menetelmille on korkeaa. Perinteisemmät raskaat ja suunnitteluvetoiset ohjelmistokehitysmenetelmät, joihin kollektiivisesti viitataan vesiputousmalleina, ovat epätehokkaita muutoksien ja epävarmuustekijöiden huomioon ottamisessa. Vastauksena vesiputousmalliin puutteisiin on kehitetty kevyempiä niin kutsuttuja ketteriä menetelmiä (eng. agile methods), joidenka tarkoituksena on helpottaa reagoimista muuttuviin vaatimuksiin ja tuottaa paremmin käyttäjien toiveisiin vastaavia ohjelmistoja. (Beck et al. [2001](#)).

Ketterä ohjelmistokehitys on saavuttanut suurta jalansijaa ohjelmistokehityksen parissa ja se on Digital.ai:n kyselyn ([2021](#)) mukaan nykyään yleisempi lähestymistapa ohjelmistokehitykseen kuin perinteinen vesiputousmalli. Ketterien kehitysmenetelmien on tutkittu johtavan vesiputousmalliin verrattuna lisääntyneeseen tehokkuuteen ja laatuun (Livermore [2008](#)), lyhyempiin tuotekehityssykleihin (Petersen ja Wohlin [2010](#)) ja lisääntyneeseen asiakas- ja ohjelmistokehittäjätyytyväisyyteen (Dybå ja Dingsøyr [2008](#)). Näiden hyötyjen vuoksi kaikenkokoiset organisaatiot ovat alkaneet kiinnostumaan ketteristä ohjelmistokehityksen menetelmistä.

Ketterät menetelmät ovat alun perin suunniteltu yksittäisten tiimien käyttöön suhteellisen pienissä ohjelmistoprojekteissa (Boehm ja Turner [2005](#)). Niihin ei täten ole määritelty toimintamalleja esimerkiksi useamman ketterän tiimin väliseen kommunikointiin tai muihin suurten ohjelmistoprojektien yleisiin haasteisiin. Näin ketterien menetelmien käyttöönotto laajan skaalan projekteissa ja isoissa organisaatioissa tuo mukanaan omat haasteensa. Laajan skaalan ketterässä ohjelmistokehityksessä useampi erillinen ketterä tiimi työskentelee yhteisen tavoitteen eteen. Erilaisia skaalattuja ketteriä ohjelmistokehitysmenetelmiä on kehitetty vastaamaan laajojen ohjelmistoprojektien haasteisiin, kuten Scaled Agile Framework (SAFe)(Leffingwell [2007](#)) ja Large-scale Scrum (LeSS) (Larman [2010](#)).

Yksi ketterien menetelmien skaalamisen haasteista on vähentynyt tiimien autonomia eli niiden heikentynyt kyky hallinnoida itseään ja päättää tehtävistään itsenäisesti. Laajan skaalan ohjelmistoprojekteilla on tarve linjata tiimit työskente-

lemään yhteisen tavoitteen eteen, ja se tarkoittaa jonkinasteista tiimien ulkopuolelta tulevaa hallintoa ja sääntelyä. Isommassa projektissa tiimeillä ei voi myöskään olla kaikkea olennaista tietoa tai osaamista täysin itsenäiseen toimintaan, ja tämä luo vääjäämättömiä riippuvuuksia muihin ketteriin tiimeihin ja ympäröivään organisaatioon.

Moe, Šmite et al. (2021) esittävät artikkelissaan aikaisempien tutkimusten perusteella, kuinka autonomian ja itsehallinnon on löydetty lisäävän luovuutta, produktiivisuutta ja kykyä innovointiin (Fenton-O’Creedy 1998; Hoegl ja Parboteeah 2006; Mathieu et al. 2008). Autonomian ja itsehallinnon on löydetty myös lisäävän työntekijöiden työmotivaatiota ja -tyytyväisyyttä. (Langfred 2000). Näiden hyötyjen nojalla olisi perusteltua pyrkiä pitämään tiimien autonomia korkealla tasolla kuitenkin sortumatta ajatukseen, jossa korkein mahdollinen autonomian taso johtaisi parhaimpiin tuloksiin. Jos jokainen tiimi työskentelee itsenäisesti, tiimien ohjelmistokehitysprosessit ja tekniset ratkaisut tulevat vääjäämättä kehittymään keskenään erilaisiksi mahdollisesti haitaten tiimien keskinäistä yhteystyötä (Moe, Šmite et al. 2021).

Tämän kirjallisuustutkimuksen tavoitteena on selvittää mitkä ovat tavallisimpia haaste- ja menestystekijöitä ketterien tiimien autonomian kannalta laajan skaalan ketterässä ohjelmistokehityksessä. Tämä tutkimus myös selvittää optimaalista suhdetta tiimien autonomian ja yhteisen linjauksen välillä etsien toimivia käytäntöjä sen saavuttamiseksi. Nämä tavoitteet voidaan muotoilla kolmeksi tutkimuskysymykseksi (TK), joihin tämä tutkimus etsii vastausta:

*TK1: Mitä haasteita liittyy tiimien autonomiaan laajan skaalan ketterässä ohjelmistokehityksessä?*

*TK2: Mitkä ovat menestystekijöitä, jotka edistävät tiimien autonomiaa laajan skaalan ketterässä ohjelmistokehityksessä?*

*TK3: Miten tiimien autonomiaa voidaan tasapainottaa tiimien ohjaamisen ja keskinäisen linjauksen kanssa?*

Tämä kirjallisuustutkimus on rakentunut seuraavasti: Luvussa 2 esitellään aiheeseen liittyvää taustaa ja termistöä. Luvussa 3 käydään läpi käytetty aineisto ja

menetelmät. Luvussa 4 esitellään tutkimuksen tulokset. Luvussa 5 pohditaan tulosten merkitystä ja muodostetaan tuloksista johtopäätökset. Tutkimuksen päättää luku 6, jossa on tutkimuksen yhteenveto.



## 2 Tausta ja termistö

Tässä luvussa laajennetaan johdannossa esiteltyä taustaa ja määritellään tässä tutkimuksessa käytettävää termistöä. Ensimmäiseksi tässä luvussa käsitellään ketterän ohjelmistokehityksen historiaa, ja kuinka se kehittyi vastauksena perinteisille suunnitteluvetoisille malleille. Toisessa alaluvussa esitellään haasteita, joita ketterien menetelmien käytöstä syntyy laajan skaalan ohjelmistokehitysprojekteissa. Kolmannessa alaluvussa määritellään autonomiset tiimit, esitellään tiimien autonomiasta koituvia hyötyjä ja syitä rajoittaa autonomiaa laajan skaalan ohjelmistokehityksessä. Neljännessä alaluvussa esitellään miten laajan skaala ymmärretään tässä tutkimuksessa. Viidennessä ja viimeisessä alaluvussa määritellään organisaatiollisen työn neljä eri funktiota, ja kerrotaan miten ne liittyvät tutkimuksen tuloksiin.

### 2.1 Ketterien menetelmien kehitys vastauksena vesiputousmallille

Ohjelmistoja on perinteisesti kehitetty suunnitteluvetoisilla menetelmillä, joissa ohjelmistokehitys rajatuu tarkasti eri vaiheisiin, joista jokainen suoritetaan loppuun ennen kuin seuraavaan voidaan siirtyä. Näitä vaiheita ovat muun muassa ohjelmistolta vaadittavien ominaisuuksien kerääminen, arkkitehtuurin suunnittelu, ohjelmiston toteutus ja ohjelmiston testaus. Näitä perinteisiä, suunnitteluvetoisia menetelmiä nimitetään kollektiivisesti vesiputousmalleiksi. Nimitys ”vesiputousmalli” juontaa siitä, että menetelmässä työprosessi ikään kuin virtaa alas tarkasti rajattuja työvaiheita pitkin vesiputouksen lailla (Sommerville 2016, s. 47–48). Vesiputousmallin tyypilliset piirteet tekevät siitä kuitenkin raskaan ja virhealttiin menetelmän, jossa on hyvin vaikea ottaa kehitysprosessin aikana ilmeneviä muutoksia huomioon (Sommerville 2016, s. 48–49). Näissä menetelmissä asiakas myös saa ohjelmiston käyttöönsä vasta aivan prosessin lopussa, ja näin voi ilmetä, että ohjelmisto ei vastaakaan täysin asiakkaan todellisia toiveita tai tarpeita.

Vesiputousmallin puutteisiin vastatakseen Beck et al. (2001) kirjoittivat ketterän manifeston, jossa määriteltiin ketterien ohjelmistokehitysmenetelmien arvot ja

prinsiipit. Näitä arvoja ovat esimerkiksi toimivan ohjelmiston tuottaminen ennen kattavaa dokumentaatiota, muutokseen reagoiminen suunnitelman seuraamisen sijaan ja yksilöiden ja interaktioiden asettaminen työkalujen ja prosessien edelle (Beck et al. 2001). Suosituimpia ketterän manifeston ajautuksiin pohjautuvia ketteriä menetelmiä ovat Scrum, Kanban, Extreme Programming (XP) ja Lean Startup (digital.ai 2021), jotka kaikki pyrkivät konkretisoimaan ketterän manifeston ajatuksia osaksi mallien käytäntöjä.

Verrattuna vesiputousmalliin ketterän ohjelmistokehityksen on todettu johtavan tehokkaampaan työprosessiin ja parempaan ohjelmiston laatuun (Livermore 2008), lyhyempiin tuotekehityssykleihin (Petersen ja Wohlin 2010) ja lisääntyneeseen asiakas- ja ohjelmistokehittäjäytytyvääisyyteen (Dybå ja Dingsøyr 2008). Näiden hyötyjen nojalla myös suuremmat organisaatiot ovat kiinnostuneet ketterien menetelmien käyttöönotosta. Ketterä ohjelmistokehitys onkin nykyään suosituimpi lähestymistapa ohjelmistokehitykseen kuin perinteinen vesiputousmalli (digital.ai 2021).

## 2.2 Ketterien menetelmien skaalauksen haasteet

Ketterät menetelmät on alun perin suunniteltu yksittäisten tiimien käyttöön verrattain pienissä ohjelmistoprojekteissa, joissa on ollut selkeä yksittäinen asiakastaho ja pääsy nopeaan palautteeseen tältä asiakkaalta (Boehm ja Turner 2005). Ketterissä menetelmissä ei täten ole määritelty toimintamalleja esimerkiksi useamman ketterän tiimin väliseen kommunikointiin tai suuren ohjelmistoprojektin moninaisten erisidosryhmien huomioon ottamiseen. Näin ketterillä menetelmillä työskentely laajan skaalan projekteissa ja isoissa organisaatioissa tuo mukanaan omat haasteensa.

Tarve organisaatioille valvoa laatua ja suorittaa toistuvia ja koordinoituja ohjelmiston julkaisuja pakottaa organisaatioita hallinnoimaan, valvomaan ja standardisoimaan monen ketterän tiimin ohjelmistokehitystä. Näin ketterien tiimien pitää laajassa skaalassa olla linjattuna muiden tiimien ja lopun organisaation kanssa. Jos ketterät tiimit rikkovat ohjelmiston laatua tai funktionaliteettia tai niiden omat työosuudet ovat myöhässä, vaikuttaa se muihin ketteriin tiimeihin ja toimitettaviin

tuotteisiin (eng. deliverables) negatiivisesti. Toisaalta tämä tarve linjata työtä, prosesseja, teknologiaa ja koordinoita vertikaalisti ylhäältä alas on uhka tiimien autonomialle, joka on avaintekijä ketteryyteen. (Moe, Dahl et al. 2019.)

Erilaisia ketterien konsulttien kehittämiä viitekehyksia (eng. framework) erityisesti laajan skaalan ketterään ohjelmistokehitykseen on noussut vastamaan näihin ongelmiin, kuten Scaled Agile Framework (SAFe) (Leffingwell 2007), Large-scale Scrum (LeSS) (Larman 2010) ja Disciplined Agile Delivery (DAD) (Ambler ja Lines 2012), joissa määritellään muun muassa se, miten useat ketterät tiimit kommunikoivat keskenään ja ympäröivän organisaation kanssa. Viitekehysistä ei kuitenkaan ole monia itsenäisiä empiirisiä tutkimuksia, jotka tutkisivat miten viitekehykset ja niiden suositukset toimivat käytännössä (Paasivaara 2017).

## 2.3 Autonomian hyödyt ja tarpeet sen rajoitukselle

Autonomisilla tiimeillä tarkoitetaan Guzzon ja Dicksonin (1996) mukaan tiimejä, joille on annettu vahva auktoriteetti ja vastuu monista heidän työnsä aspekteista, kuten työn suunnittelusta, aikataulutuksesta, tehtävien jaosta tiimin jäsenten kesken ja taloudellisiin seurauksiin johtavien päätösten teosta. Näin tiimien autonomian tasolla voidaan mieltää sitä vastuun ja auktoriteetin määrää, joka tiimeille annetaan esimerkiksi edellä mainittuihin toimintoihin. Hackmanin (1986) teorian perspektiivistä tiimien autonomian tasolla voidaan viitata siihen, kuinka paljon valtaa tiimeillä on itse toteuttaa Hackmanin (1986) neljää työn tekemisen funktiota, joista kerrotaan lisää luvussa 2.5.

Moe, Šmite et al. (2021) esittävät artikkelissaan aikaisempien tutkimusten perusteella, kuinka autonomian ja itsehallinnon on löydetty lisäävän työntekijöiden luovuutta, produktiivisuutta ja kykyä innovointiin (Fenton-O’Creevy 1998; Hoegl ja Parboteeah 2006; Mathieu et al. 2008). Autonomian on näytetty myös lisäävän työntekijöiden motivaatiota ja tyytyväisyyttä työhön (Langfred 2000). Autonomia on myöskin yksi Ryan and Decin (2012) kolmesta itseohjautuvuusteorian tunnistamasta psykologisesta perustarpeesta, jotka yhdessä selittävät ihmisten hyvinvointia ja halua omaehtoiseen toimintaan.

Näiden hyötyjen valossa voisi ajatella, että ketterien tiimien autonomia tulisi pyrkiä maksimoimaan laajan skaalan ketterässä ohjelmistokehityksessä, mutta tätä ideaa vastaan löytyy todistusaineistoa. Dikertin et al. (2016) meta-analyysissä löydettiin, että tiimien autonomia oli sekä menestys- että haastetekijä laajan skaalan ketterässä ohjelmistokehityksessä. Itsenäisesti työskentelevät ketterät tiimit esimerkiksi keskittyivät vain omiin tavoitteisiinsa jättäen yhteiset tavoitteet vähemmälle huomiolle tai vaikeuttivat kommunikaatiota sivuuttaessaan laajemman projektin tai organisaation kontekstin (Dikert et al. 2016). Tiimien täysi autonomia on laajan skaalan organisaatioissa epärealistinen tavoite, koska tiimit eivät tuota lopullista työtään itsenäisesti. Tiimien linjausta tarvitaan laadun varmistamiseksi ja keskinäisten riippuvuuksien hallinnoimiseksi. (Olsson Holmström ja Bosch 2016; Moe, Šmite et al. 2021.)

Näin ollen laajan skaalan ketterässä ohjelmistokehityksessä tiimien autonomialla voidaan ajatella olevan optimaalinen taso, jonka puitteissa tiimit on linjattu tarpeeksi vahvasti työskentelemaan yhteisten tavoitteiden eteen keskenään yhteensopivilla tavoilla ilman, että linjaus kuihduttaisi liiaksi autonomiasta koituvia hyötyjä. Eli kyky tasapainottaa hajautettu päätöksentekoprosessi keskitetyn päätöksentekorakenteen kanssa on tärkeä menestystekijä laajan skaalan ohjelmistokehitysprojekteissa (Dingsøyr et al. 2018).

## 2.4 Laajan skaalan määritelmä

Seuraten Uludagin et al. (2020) määritelmää, tässä tutkimuksessa laajan skaalan ketterä ohjelmistokehitys ymmärretään kahdeksi ketterän ohjelmistokehityksen muodoksi: Ensinnäkin se ymmärretään ketteräksi kehitykseksi, jossa vähintään kaksi erillistä ketterää tiimiä työskentelee yhteisen tavoitteen eteen. Tätä voidaan kutsua laajaksi monen tiimin ketteryydeksi (eng. large agile multi-team setting). Toiseksi se ymmärretään ketteräksi kehitykseksi, jossa useampi vähintään kahden ketterän tiimin ryhmittymä työskentelee saman organisaation sisällä. Tätä voidaan kutsua organisaatiolliseksi ketteryydeksi (eng. organizational agility). Organisaatiolliseen ketteryyteen liittyy myös ketterien periaatteiden käyttö kokonaisvaltaisesti organisaatiossa.

## 2.5 Organisaatiollisen työn neljä eri funktiota

Hackmanin (1986) mukaan neljä erilaista funktiota täytyy toteuttaa, kun töitä tehdään organisaatiossa:

- (1) Jonkun pitää asettaa suunta organisaatiolliselle työlle, joka pitää tehdä.
- (2) Jonkun pitää suunnitella työn toteuttava yksikkö ja saada työn tekemiselle organisaatiollinen tuki.
- (3) Jonkun pitää hahmottaa, monitoroida ja hallinnoida työprosessia.
- (4) Jonkun pitää tehdä itse työ.

Itsehallinnoituvat (eng. self-managing) työyksiköt voidaan Hackmanin (1986) mukaan määritellä sen perusteella miten auktoriteetti jakautuu näiden neljän funktion sisällä. Hackman (1986) jakaa työyksiköt neljään eri tyyppiin riippuen siitä kuinka paljon auktoriteettia niillä on neljässä eri työn tekemisen funktiossa. Tämän tutkimuksen kontekstissa itsehallinnoituvaksi työyksiköksi voidaan ymmärtää organisaation sisällä toimiva ketterä tiimi.

Hackmanin (1986) neljä eri funktiota auttavat tässä tutkimuksessa hahmottamaan mihin työn tekemisen osa-alueeseen tai -alueisiin tuloksissa löydetty haaste- ja menestystekijät vaikuttavat. Näin Hackmanin (1986) funktiot auttavat myös suodattamaan pois niitä haaste- ja menestystekijöitä, jotka eivät spesifisti liity tiimien autonomiaan tai päätösvallan jakautumiseen työn tekemisen eri alueilla laajan skaalan ketterässä ohjelmistokehityksessä. Haaste- ja menestystekijöiden linkittäminen Hackmanin (1986) funktioihin antavat myös näkymän siihen, millä funktion tasoilla kohdataan eniten haaste- ja menestystekijöitä tässä tutkimuksessa tarkastellun aineiston pohjalta.

Moe, Šmite et al. (2021) ovat aiemmin soveltaneet Hackmanin (1986) teoriaa tutkimuksessaan auktoriteetin jakautumisesta laajan skaalan ketterässä ohjelmistokehityksessä.

### 3 Aineisto ja menetelmät

Tässä luvussa käsitellään ensin miten tutkimuksen aineisto valittiin. Toiseksi luvussa kerrotaan kuinka tutkimuksen tulokset muodostettiin.

Aineiston haun pohjana toimi Uludagin et al. (2020) kartoitustutkimus, joka kartoitti vuosina 2001–2019 julkaistut laajan skaalan ketterää ohjelmistokehitystä koskevat vertaisarvioidut tutkimukset. Uludag et al. (2020) jakoivat laajan skaalan ketterän ohjelmistokehityksen tutkimuskentän eri tutkimussuuntauksiksi, joista yksi oli tiimien autonomian tutkimus. Tiimien autonomian tutkimussuuntaukseen luokitellut tutkimukset otettiin osaksi tämän kirjallisuustutkimuksen aineistoa. Uludagin et al. (2020) tiimien autonomian tutkimussuuntauksen pohjalta saatiin seitsemän tutkimusta osaksi tämän kirjallisuustutkimuksen aineistoa.

Uludagin et al. (2020) systemaattinen kartoitustutkimus rajoittui tutkimuksiin, jotka oli julkaistu vuosina 2001–2019. Täten oli tarpeellista tehdä tietokantahaku vuosina 2020–2021 julkaistuille artikkeleille. Vuosi 2022 jätettiin pois, koska tätä kirjallisuustutkimusta aloitettiin kirjoittamaan tammikuussa 2022. On perusteltua rajata ennen vuotta 2001 julkaistut artikkelit pois, koska vuonna 2001 julkaistiin ketterä manifesti (Beck et al. 2001), jonka avulla ensimmäisen kerran määriteltiin ketterät menetelmät niin kuin ne tässä kirjallisuustutkimuksessa ymmärretään.

Tämän kirjallisuustutkimuksen tietokantahaku perustuu Uludagin et al. (2020) tutkimuksessa käytetyille tietokannoille ja hakutermistöön. Verrattuna Uludagin et al. (2020) tutkimuksessa käytettyihin tietokantoihin, tässä tutkimuksessa ei käytetty tietokantaa Science Direct sen kahdeksan loogisen operaattorin rajoituksen vuoksi. Aineiston keruussa käytetyt tietokannat on esitetty taulukossa 1. Koska tämä kirjallisuustutkimus keskittyy erityisesti tiimien autonomiaan, lisättiin Uludagin et al. (2020) hakusanoihin taulukossa 2 esitetty hakusanojen kokoelma ”tiimien autonomia”. Tämä hakusanojen kokoelma koottiin Uludagin et al. (2020) tutkimussuunnan ”tiimien autonomia” artikkeleista löytyvien avainsanojen (eng. keywords) pohjalta. Tietokantahaussa käytetyt hakutermit löytyvät taulukosta 2. Lopullinen tietokantahaku tehtiin yhdistelemällä hakutermien kokoelmat seuraavasti:

(Ketterä ohjelmistokehitys AND (Laajan skaalan kehitys  
OR Skaalausmallit)) AND Tiimien autonomia.

Tietokanta	Internet-sivu
IEEE Xplore	<a href="http://ieeexplore.ieee.org/">http://ieeexplore.ieee.org/</a>
ACM Digital Library	<a href="http://dl.acm.org/">http://dl.acm.org/</a>
Web of Science	<a href="https://www.webofknowledge.com/">https://www.webofknowledge.com/</a>
Scopus	<a href="https://www.scopus.com/home.uri">https://www.scopus.com/home.uri</a>
AIS eLibrary	<a href="https://aisel.aisnet.org/">https://aisel.aisnet.org/</a>

Taulukko 1: Tutkimuksessa käytetyt tietokannat

Tietokantahausta valittiin tutkimukset osaksi aineistoa seuraavasti: Ensinnäkin hausta rajattiin tutkimukset sen pohjalta liittyivätkö ne laajan skaalan ketterään ohjelmistokehitykseen tutkimusten otsikoiden, avainsanojen ja tarvittaessa tiivistelmän perusteella. Toiseksi rajattiin pois ne laajan skaalan ketterän ohjelmistokehityksen tutkimukset, jotka eivät pääasiallisesti tai suuressa määrin tarkastelleet laajan skaalan ketterää ohjelmistokehitystä tiimien autonomian perspektiivistä tutkimusten avainsanojen, tiivistelmän ja tarvittaessa kokotekstin perusteella. Tietokantahaun pohjalta sisällytettiin kolme tutkimusta osaksi tämän kirjallisuustutkimuksen aineistoa.

Kirjallisuustutkimukseen valittu aineisto on esitetty taulukossa 3. Aineiston artikkeleille on annettu koodit luvussa 4 tulevaa haaste- ja menestystekijöiden taulukoinnista varten eli taulukoita 4 ja 6 varten.

Aineistoon valittujen tutkimusten pohjalta kirjattiin haaste- ja menestystekijöitä tiimien autonomialle, jonka jälkeen kirjatut haaste- ja menestystekijät jaoteltiin niiden pohjalta muodostuneiden teemojen alle. Teeman muodostumisen ehtona oli, että vähintään kahdessa tutkimuksessa kohdattiin siihen liittyviä haaste- tai menestystekijöitä. Jokainen teema linkitettiin Hackmanin (1986) organisaatiollisen työn funktioihin. Täten haaste- ja menestystekijöiden teemoille oli selvät organisaatiollisen työn osa-alueet, jossa ne joko heikensivät tai lisäsivät tiimien auktoriteettia. Teemojen linkityksellä Hackmanin (1986) funktioihin saatiin myös suodatettua pois ne haaste- ja menestystekijät, jotka eivät liittyneet tiimien autonomiaan. Esimerkiksi eräässä aineiston tutkimuksista demopäivien koettiin edistävän tiimien

Termien kokoelma	Hakutermi
Ketterä ohjelmistokehitys	(agile <b>OR</b> agility <b>OR</b> extreme programming <b>OR</b> XP <b>OR</b> feature driven development <b>OR</b> FDD <b>OR</b> scrum <b>OR</b> crystal <b>OR</b> pair programming <b>OR</b> test-driven development <b>OR</b> TDD <b>OR</b> leanness <b>OR</b> lean software development <b>OR</b> lean development <b>OR</b> LSD) <b>AND NOT</b> manufacturing
Laajan skaalan kehitys	large-scale <b>OR</b> scaling
Skaalausmallit	Crystal Family <b>OR</b> Dynamic Systems Development Method Agile Project Framework for Scrum <b>OR</b> Scrum of Scrums <b>OR</b> Enterprise Scrum <b>OR</b> Agile Software Solution Framework <b>OR</b> Large-Scale Scrum <b>OR</b> Scaled Agile Framework <b>OR</b> Disciplined Agile <b>OR</b> Spotify Model <b>OR</b> Mega Framework <b>OR</b> Enterprise Agile Delivery and Agile Governance Practice <b>OR</b> Recipes for Agile Governance in the Enterprise <b>OR</b> Continuous Agile Framework <b>OR</b> Scrum at Scale <b>OR</b> Enterprise Transition Framework <b>OR</b> ScALeD Agile Lean Development <b>OR</b> eXponential Simple Continuous Autonomous Learning Ecosystem <b>OR</b> Lean Enterprise Agile Framework <b>OR</b> Nexus <b>OR</b> FAST Agile
Tiimien autonomia	Autonomous teams <b>OR</b> Team autonomy <b>OR</b> Autonomy <b>OR</b> Self-management <b>OR</b> Self-organizing teams

Taulukko 2: Tietokantahaussa käytetyt hakutermin kokoelmat

välistä kommunikaatiota. Tämä menestystekijä ei kuitenkaan edistänyt tiimien auktoriteettia millään Hackmanin (1986) funktion osa-alueella, ja näin ollen kyseinen menestystekijä voitiin jättää tulosten ulkopuolelle.



Artikkeli	Lähde	Koodi
Mikalsen et al. <a href="#">2019</a>	Uludag et al. <a href="#">2020</a>	T1
Šablīs ja Šmite <a href="#">2016</a>	Uludag et al. <a href="#">2020</a>	T2
Petit ja Marnewick <a href="#">2019</a>	Uludag et al. <a href="#">2020</a>	T3
Gundelsby <a href="#">2018</a>	Uludag et al. <a href="#">2020</a>	T4
Tendedez et al. <a href="#">2018</a>	Uludag et al. <a href="#">2020</a>	T5
Moe, Dahl et al. <a href="#">2019</a>	Uludag et al. <a href="#">2020</a>	T6
Gustavsson <a href="#">2019</a>	Uludag et al. <a href="#">2020</a>	T7
Moe, Šmite et al. <a href="#">2021</a>	Tietokantahaku	T8
Mohagheghi et al. <a href="#">2020</a>	Tietokantahaku	T9
Salameh ja Bass <a href="#">2020</a>	Tietokantahaku	T10

Taulukko 3: Tutkimuksessa käytetyt artikkelit

## 4 Tulokset

Tässä luvussa vastataan ensimmäiseksi tutkimuksen ensimmäiseen tutkimuskysymykseen; *mitä haasteita liittyy tiimien autonomiaan laajan skaalan ketterässä ohjelmistokehityksessä?* Toiseksi luvussa vastataan tutkimuksen toiseen tutkimuskysymykseen; *mitkä ovat menestystekijöitä, jotka edistävät tiimien autonomiaa laajan skaalan ketterässä ohjelmistokehityksessä?*

### 4.1 Haasteet

Tämä alaluku käsittelee haastetekijät teemoittain vastaten tutkimuksen ensimmäiseen tutkimuskysymykseen. Haasteet teemoittain on esitelty taulukossa 4. Taulukossa on esitelty myös missä tutkimuksissa kutakin haasteteemaa esiintyi. Taulukossa 5 on esitelty mihin Hackmanin ([1986](#)) organisaatiollisen työn funktioihin haasteteemat linkittyvät.

***Ei tarvittavaa osaamista tai resursseja:*** Tarvittavan osaamisen ja organisaatiollisen tuen puuttuminen vähensivät tiimien kykyä itsenäiseen työhön. Tämä haasteteema liittyy Hackmanin ([1986](#)) funktioihin (2) ja (4), koska haasteet vähensivät tiimien auktoriteettia saada työn tekemiseen organisaatiollista tukea tai tehdä itse työtä.

Haasteteema	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Ei tarvittavaa osaamista tai resursseja	X	X					X		X	
Ulkoa tulevat häiriötekijät	X				X	X				
Ulkoa tulevat linjaukset työhön ja tiimin rakenteeseen					X	X	X	X		
Vaikeus rajata tiimien työtä itsenäisiksi osiksi				X	X	X			X	X
Vähäinen luottamus ja valtuutukset päätöksiin			X		X					X
Epäselvyys yhteisistä tavoitteista ja rooleista						X	X			

Taulukko 4: Haasteteemat tutkimuksittain

Haasteteema	Hackmanin (1986) funktiot
Ei tarvittavaa osaamista tai resursseja	(2) ja (4)
Ulkoa tulevat häiriötekijät	(3) ja (4)
Ulkoa tulevat linjaukset työhön ja tiimin rakenteeseen	(1), (2), (3) ja (4)
Vaikeus rajata tiimien työtä itsenäisiksi osiksi	(3)
Vähäinen luottamus ja valtuutukset päätöksiin	(3) ja (4)
Epäselvyys yhteisistä tavoitteista ja rooleista	(1) ja (2)

Taulukko 5: Haasteteemat linkitettyinä Hackmanin (1986) organisaatiollisen työn funktioihin

Monella organisaatiolla oli yksinkertaisesti vaikeuksia saada tarvittavaa osaamista ketteriin tiimeihin, jotta ne pystyisivät työskentelemään itsenäisesti (Šāblis ja Šmite 2016; Gustavsson 2019; Mohagheghi et al. 2020). Tarvittavan osaamisen puuttuminen ja organisaatiollisen tuen puute yhdistyivät esimerkiksi tilanteissa, joissa tiimi oli riippuvainen organisaatiossa työskentelevästä tiimin ulkoisen asiantuntijan osaamisesta, joka ei ollut sitoutunut ketterien periaatteiden toimintaan tai tiimin tavoitteiden edistämiseen (Mikalsen et al. 2019). Vähäiset resurssit näkyivät esimerkiksi tapauksessa, jossa tiimin jäsenet joutuivat jakamaan aikaansa tiimin omien tavoitteiden ja tiimin ulkopuolisten työtehtävien välillä ilman, että tiimillä itsellään oli sananvaltaa työajan käyttöön (Mikalsen et al. 2019). Tarvittavan osaamisen

puuttuessa ketterät tiimit joutuivat nojaamaan esimerkiksi tiimin ulkopuolisiin asiantuntijoihin. Näihin asiantuntijoihin oli ajoittain vaikea saada yhteys, jos he työskentelivät etänä suhteessa tiimeihin. (Šāblis ja Šmite 2016.) Osassa tapauksista riippuvuudet ulkoisen organisaation tukeen hidastivat tiimien työskentelyä ja haittasivat niiden kykyä suorittaa tehtäviä itsenäisesti (Mikalsen et al. 2019; Petit ja Marnewick 2019). Tiimeillä ei myöskään aina ollut valtaa vaikuttaa ulkoiseen organisaatioon, jotta organisaatiolta olisi saatu itsenäiseen työhön vaadittavia resursseja (Mikalsen et al. 2019). Myös organisaation rahoitusmalli loi haasteita saada tarpeeksi resursseja tiimien itsenäiselle työskentelylle (Mohagheghi et al. 2020).

***Ulkoa tulevat häiriötekijät:*** Tiimien ulkopuolelta tulevat häiriötekijät haittasivat tiimien itsenäistä työskentelyä. Tämä haasteteema vei valtaa tiimeiltä erityisesti Hackmanin (1986) funktioiden (3) ja (4) alueilla, sillä ulkoa tulevat häiriötekijät tekivät työprosessien itsenäisestä hallinnoimisesta haastavaa ja vähensivät tiimien päättäväisyyttä työskennellä itse haluamallaan tavalla esimerkiksi hyödyntäen syvää keskittymistä ohjelmiston toteutusvaiheessa.

Yksi häiriötekijöistä oli ulkopuolelta jatkuvasti virrannut suuri informaation määrä, josta pääosan ei koettu koskevan ketteriä tiimejä (Moe, Dahl et al. 2019). Ylhäältä tiimeille annetut yllättävät ja akuutit tehtävät haittasivat tiimien kykyä itsenäisesti suunnitella ja toteuttaa työtään. Yllättävien ja akuuttien tehtävien seurauksena tiimit joutuivat jättämään omat sprintin aikaiset tehtävänsä ja ottamaan hoidettavakseen kehitysjonon (eng. backlog) ulkopuolelta tulevia tehtäviä. (Mikalsen et al. 2019; Moe, Dahl et al. 2019.) Myös ulkoa määrättyjen kokousten suuri määrä haittasi tiimien kykyä itsenäisesti suunnitella ajankäyttöään ja suorittaa tehtäviään varsinkin, kun tapaamisten pääasiallisena tarkoituksena oli raportoida tiimien edistymistä johdolle (Tendedez et al. 2018).

***Ulkoa tulevat linjaukset työhön ja tiimin rakenteeseen:*** Tiimien kykyä itsenäisesti tehdä työtä haittasi tiimien ulkopuolelta tulevat linjaukset itse työn tekoon ja monitorointiin, tiimien tavoitteisiin ja tiimin rakenteeseen. Tämä haasteteema liittyy kaikkiin neljään Hackmanin (1986) organisaatiollisen työn funktioon, sillä teeman haasteet veivät valtaa tiimeiltä tehdä työtä ja monitoroida sitä itse

parhaaksi näkemällään tavalla, asettaa työlleen suunta tavoitteiden muodossa, ja suunnitella itsenäisesti työn toteuttavaa yksikköä.

Vaikka tiimit olisivatkin lähtökohtaisesti saaneet päättää miten lähestyä työtehtäviään, heille asetettiin johdon puolelta useita pakollisia prosesseja ja käytäntöjä, joilla työn tekoa ja hallinointia tuli suorittaa (Tendedez et al. 2018; Moe, Šmite et al. 2021). Osassa tapauksista tiimit saivat tavoitteet ja vaatimukset työlleen suoraan ylempää ilman mahdollisuutta vaikuttaa niiden asetteluun (Moe, Dahl et al. 2019; Mikalsen et al. 2019). Myös tiimien rakennetta muutettiin ilman konsultointia tiimeiltä, joka haittasi tiimien kykyä suunnitella työn toteuttavaa yksikköä (Moe, Šmite et al. 2021). Monessa tapauksessa tiimien työtehtävät tulivat pääosin ulkoa ilman, että tiimit itse olisivat päässeet vaikuttamaan niiden valintaan (Moe, Dahl et al. 2019; Gustavsson 2019; Moe, Šmite et al. 2021). Työtehtäviä annettiin tiimeille ylempää esimerkiksi perustuen tiimien aiempiin työtehtäviin, mikä ei edistänyt tiimien laaja-alaisen osaamisen kehittymistä (Moe, Šmite et al. 2021).

***Vaikeus rajata tiimien työtä itsenäisiksi osiksi:*** Tiimien autonomiaa haittasi vaikeus rajata tiimien työ- ja vastuualueita itsenäisiksi kokonaisuuksiksi, jotka eivät olisi riippuvaisia muista tiimeistä tai ympäröivästä organisaatiosta. Tämä haasteteema liittyy Hackmanin funktioon (3), koska selkeästi rajatun vastuualueen puute teki tiimien työprosessin itsenäisestä hahmottamisesta ja hallinnoimisesta erittäin haasteellista.

Huonosti rajatuista työalueista seuranneet tiimien väliset riippuvuudet esimerkiksi hidastivat ketterien tiimien työskentelyä ja vähensivät niiden kykyä itsenäisesti edistää työtään (Tendedez et al. 2018; Moe, Dahl et al. 2019). Ohjelmiston arkkitehtuuri oli yksi tekijöistä, joka vaikeutti työn rajaamista itsenäisiin osiin. Arkkitehtuuri loi osassa tapauksista riippuvuuksia muihin tiimeihin, ja arkkitehtonisten päätösten linjaus ja hallinnointi autonomisten tiimien kesken koettiin haastavaksi. (Gundelsby 2018; Salameh ja Bass 2020.) Toisessa tapauksessa yhtenäisten tuoteportfolioiden määrittely ketterille tiimeille ja riippuvuuksien pitäminen vähäisinä muihin tiimeihin koettiin haasteelliseksi (Mohagheghi et al. 2020).

***Vähäinen luottamus ja valtuutukset päätöksiin:*** Vähäinen luottamus ketteriin tiimeihin, ja niiden rajalliset mahdollisuudet tehdä itse päätöksiä haittasivat

tiimien itsenäistä työskentelyä. Vähäinen luottamus tiimeihin vähensi erityisesti tiimien auktoriteettia itse monitoroida ja hallinnoida omaa työprosessiaan vaikuttaen Hackmanin funktioon (3). Vähäiset valtuutukset päätöksiin taas heikensivät tiimien valtaa tehdä itse työn tekemistä koskevia päätöksiä vaikuttaen Hackmanin funktioon (4).

Luottamuksen puute näkyi esimerkiksi siten, että johto ei täysin hyväksynyt sitä minkä asioiden kanssa tiimit työskentelivät. Tiimit myös kokivat painetta johdolta merkitä työtehtäviä tehdyksi. Tämä vähensi tiimien vapautta itse valita ja hallinnoida työtehtäviään niin kuin tiimi itse parhaaksi näkee. (Tendedez et al. 2018.)

Toinen tapa, jolla vähäinen luottamus ja valtuutukset päätöksiin näkyivät oli byrokraattisten ja hierarkkisten julkaisukäytäntöjen muodossa. Tiimien piti eräässä tapauksessa hyväksyttää kaikki tekemänsä muutokset ohjelmistoon ryhmällä auktoriteetteja ennen kuin kyseiset muutokset voitiin integroida osaksi tuotantoa (eng. production). Auktoriteettiryhmän hidas päätöstahti ja sen luoma säännöstö kannusti tiimejä lähestymään työtään organisaation kannalta haitallisin tavoin. (Petit ja Marnewick 2019.)

Vähäiset valtuutukset tehdä itsenäisiä päätöksiä ohjelmiston arkkitehtuurista johti siihen, että tiimit olivat vahvasti riippuvaisia ulkoisesta auktoriteetista, jonka vastuulla oli hyväksyttää tiimien ehdottamat muutokset. Tästä auktoriteetista muodostui pullonkaula tiimien työlle, kun hän ylikuormittui ollessaan vastuussa usean ketterän tiimin arkkitehtuuria koskevista päätöksistä. (Salameh ja Bass 2020.)

***Epäselvyys yhteisistä tavoitteista ja rooleista:*** Epäselvyys tiimien keskinäisistä yhteisistä tavoitteista ja rooleista haittasi tiimien itsenäistä työskentelyä, koska heiltä puuttui suunta organisaatiolliselle työlle. Tiimeillä ei ollut itsellään auktoriteettia asettaa omia tavoitteita ja rooleja, koska ne tulivat tiimeille ylempää linkittäen haasteteeman Hackmanin funktioihin (1) ja (2). Kun näitä tavoitteita ei kommunikoitu tiimeille tarpeeksi hyvin tai varmistettu, että ylhäältä tulleet tavoitteet ja roolit olivat omaksuttu ketterissä tiimeissä, vaikeutti haasteteema ketterien tiimien itsenäistä työskentelyä (Moe, Dahl et al. 2019; Gustavsson 2019).

Tiimien oli vaikea rajata vastualueitaan ja lähteä työskentelemään itsenäisesti niiden eteen, kun tiimit eivät olleet selvillä siitä mitkä tehtävät olivat kenenkin vastuulla (Moe, Dahl et al. 2019; Gustavsson 2019) tai siitä mitä heidän tulisi tiiminsä kesken saavuttaa (Moe, Dahl et al. 2019). Epäselvyys tavoitteista juonsi esimerkiksi siitä, että tavoitteet määriteltiin ylempää ilman dialogia tiimien kanssa, jonka jälkeen johto ei käyttänyt tarpeeksi aikaa tai resursseja tavoitteiden kommunikoinniseen tiimeille (Moe, Dahl et al. 2019).

## 4.2 Menestystekijät

Tämä alaluku käsittelee menestystekijät teemoittain vastaten tutkimuksen toiseen tutkimuskysymykseen. Menestystekijät teemoittain on esitelty taulukossa 6. Taulukossa on esitelty myös missä tutkimuksissa kutakin menestystekijäteemaa esiintyi. Taulukossa 7 on esitelty mihin Hackmanin (1986) organisaatiollisen työn funktioihin menestystekijäteemat linkittyivät.

Menestystekijäteema	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Yhteisöllinen suunnittelu ja päätöksenteko						X	X	X		X
Työn rajaaminen itsenäisiksi kokonaisuuksiksi				X					X	
Luottamus tiimeihin ja valtuudet tehdä päätöksiä	X		X	X				X	X	X
Autonomiia edistävät käytännöt				X	X	X		X	X	X

Taulukko 6: Menestystekijäteemat tutkimuksittain

***Yhteisöllinen suunnittelu ja päätöksenteko:*** Työtä suorittavien ketterien tiimien keskinäinen työn suunnittelu ja päätöksenteko oli yksi tiimien autonomiia edistävästä menestystekijöistä. Yhteisöllinen suunnittelu ja päätöksenteko antoivat tiimeille lisääntyntä valtaa päättää keskenään työssä käytettävistä käytännöistä ja prosesseista linkittäen sen Hackmanin funktioon (4). Se antoi tiimeille myös paremmin valtaa hahmottaa, monitoroida ja hallinnoida tiimien työprosessia linkittäen sen funktioon (3).

Menestystekijäteema	Hackmanin (1986) funktiot
Yhteisöllinen suunnittelu ja päätöksenteko	(3) ja (4)
Työn rajaaminen itsenäisiksi kokonaisuuksiksi	(3)
Luottamus tiimeihin ja valtuudet tehdä päätöksiä	(3) ja (4)
Autonomiaa edistävät käytännöt	(1), (2), (3) ja (4)

Taulukko 7: Menestystekijäteemat linkitettyinä Hackmanin (1986) organisaatiollisen työn funktioihin

Yhteisöllisellä päätöksenteolla tiimit saivat enemmän valtaa päättää keskenään parhaista tavoista lähestyä työtehtäviä. Verrattuna ylempää tuleviin pakollisiin työn käytäntöihin ja prosesseihin tiimit kokivat yhteisön kesken tehtyjen päätöksien paremmin refleктоivan työn sen hetkisiä tarpeita. Yhteisöllisellä päätöksenteolla esimerkiksi arkkitehtuurista tiimit pystyivät tekemään paremmin informoituja päätöksiä, kun he oppivat käytännöstä ja sen ongelmista. Tämä myös lisäsi tiimien kykyä itsenäiseen päästä päähän (eng. end-to-end) tuotekehitykseen, kun arkkitehtuurinen osaaminen oli tiimien sisällä. (Moe, Šmite et al. 2021.)

Yhteisöllisellä päätöksenteolla arkkitehtuurista tiimit pystyvät esimerkiksi paremmin linjaamaan tiimien kesken arkkitehtonisia päätöksiä. Yhteisöllinen päätöksenteko arkkitehtuurista siirsi valtaa arkkitehtonisista päätöksistä tiimien ulkopuolisilta asiantuntijoilta tiimien sisään. (Salameh ja Bass 2020.)

Yhteisellä työn suunnittelulla ja työmäärän jakamisella tiimit pystyivät paremmin vaikuttamaan tulevan sprintin työmäärään ja jakamaan työtaakkaa tasaisemmin tiimien kesken. Tämä myös edisti tiimien välistä keskinäistä auttamista, koska tiimit olivat paremmin tietoisia toistensa vastuualueista. Näin tiimit pystyivät nopeammin ratkomaan mahdollisia keskinäisiä riippuvuuksia tukemalla sitä tiimiä työpanoksellaan, jonka työstä oli muodostunut pullonkaula muille. (Gustavsson 2019.)

Myöskin yhteisöllisesti muodostettu ymmärrys siitä mitä pitää tehdä ja missä järjestyksessä auttoi tiimejä suunnittelemaan ja toteuttamaan työtään itsenäisesti (Moe, Dahl et al. 2019).

***Työn rajaaminen itsenäisiksi kokonaisuuksiksi:*** Työn rajaaminen itsenäi-

siksi kokonaisuuksiksi auttoi tiimejä toteuttamaan työtään autonomisesti vähentämällä riippuvuuksia muihin tiimeihin ja ympäröivään organisaatioon. Tämä teema linkittyy Hackmanin funktioon (3), koska menestystekijä antoi lisääntyntä valtaa tiimeille hahmottaa ja hallinnoida työprosessejaan itsenäisesti.

Esimerkiksi ohjelmiston arkkitehtuurin rajaaminen keskinäisesti itsenäisiin osiin, jotka kommunikoivat keskenään selvästi määriteltyjen ohjelmistorajapintojen (eng. application programming interface, API) avulla edisti tiimien kykyä toimia itsenäisesti (Gundelsby 2018). Myös työn rajaaminen tiimeille tarkasti määritellyn tuoteportfolion ympärille edisti tiimien kykyä toimia autonomisesti ilman laajoja riippuvuuksia muihin tiimeihin (Mohagheghi et al. 2020).

***Luottamus tiimeihin ja valtuudet tehdä päätöksiä:*** Lisääntynyt luottamus tiimeihin, ja niille annetut valtuudet tehdä itsenäisiä päätöksiä nosti tiimien autonomian tasoa. Tiimeille annettu valta lähestyä esimerkiksi työtehtävien suunnittelua, toteutusta, ja testausta itsenäisesti oli yksi autonomian edellytyksistä tiimeille. Teema linkittyy Hackmanin funktioon (3), koska lisääntynyt luottamus edisti tiimien auktoriteettia itsenäisesti hallinnoida työprosessejaan, ja funktioon (4), koska tiimeille annettiin lisääntyntä valtaa tehdä itsenäisiä päätöksiä koskien itse työn tekoa.

Vapaus ja valtuudet tehdä työtä koskevia päätöksiä lisäsi tiimien valtaa hallinnoida ja suorittaa työtään itsenäisesti (Gundelsby 2018; Moe, Šmite et al. 2021). Toinen luottamus johdolta tiimeihin mahdollisti tiimeille sen, että he itse saivat päättää mihin työssään keskittyä ja miten lähestyä työtehtäviään (Mikalsen et al. 2019). Luottamus tiimien ja ympäröivän organisaation välillä ja valtuutukset tehdä tuotetta ja työprosesseja koskevia päätöksiä koettiin eräässä tapauksessa olevan ehtona tiimien autonomian toteutumiselle (Mohagheghi et al. 2020).

Esimerkiksi tiimeille annetut valtuudet tehdä aiempaa itsenäisempiä päätöksiä arkkitehtuurista yhteistyössä muiden tiimien kanssa lisäsi tiimien valtuuksia edistää itse työtään. Aiemmin päätökset arkkitehtuurista oli tehnyt tiimien ulkopuolinen asiantuntija, joka usein oli pullonkaulana ketterien tiimien työn edistymiselle (Salameh ja Bass 2020).



Eräässä tapauksessa tiimeille annettiin lisääntyneesti luottamusta ja valtuutuksia tehdä itse muutoksia suoraan tuotantoon (eng. production) perustuen tiimien edellisiin suorituksiin; muun muassa jokainen onnistunut muutoksien integroiminen tuotantoon osaltaan lisäsi luottamusta ja valtuutuksia, kun taas epäonnistumiset laskivat niitä. Näin tiimeille annettiin selkeä väylä, jonka avulla he pystyivät nostamaan saamaansa luottamusta ja auktoriteettia kuitenkin tasapainottaen niitä laadun hallinnan kanssa. (Petit ja Marnewick 2019.)

***Autonomiia edistävät käytännöt:*** Aineistosta nousi esiin monia yksittäisiä autonomiia edistäviä käytäntöjä, jotka yhdessä kattoivat kaikki Hackmanin työn tekemisen funktiot. Linkitetyt funktiot on esitelty käytäntöjen yhteydessä.

Tiimien konsultointi mitä tulee uusiin jäsenien rekrytointipäätöksiin ja sen suhteen, ketä uusia jäseniä tiimeihin tulee lisäsi tiimien päätäntävaltaa työyksikön suunnittelussa (Moe, Šmite et al. 2021). Tämä antoi tiimeille lisääntyntä valtaa erityisesti Hackmanin funktiossa (2), koska näin tiimeillä oli lisääntyntä auktoriteettia suunnitella työn toteuttavaa yksikköä.

Ketterän työskentelyn asenne tiimien sisällä ja ulkoisessa organisaatiossa vähensi ulkoisten riippuvuuksien vaikutusta tiimeihin ja tuki tiimien autonomiia kunnioitettava asennetta (Mohagheghi et al. 2020). Menestystekijä lisäsi tiimien valtaa Hackmanin funktiossa (2), koska tiimeillä oli tämän avulla lisääntyneet valtuudet saada työlleen muun organisaation tuki.

Resurssipohjainen rahoitusmalli, jossa ketterä tiimi sai jatkuvaa rahoitusta kattamaan tiimin työntekijöiden palkat edisti tiimin autonomiia verrattuna käytäntöön, jossa se haki rahoitusta ohjelmistoon tulevia muutoksia varten. Se antoi tapauksen tiimille enemmän valtaa suunnitella työtään pidempi tähtäin mielessä ja tasapainotamaan funktionaalisten ja teknisten muutosten välillä. Näin tiimi pystyi tarvittaessa keskittymään teknisen velan poistamiseen uusien ohjelmiston ominaisuuksien kehittämisen sijaan. (Mohagheghi et al. 2020.) Menestystekijä edisti tiimien valtaa Hackmanin funktiossa (2), koska rahoitusmalli edisti tiimien auktoriteettia saada työlleen organisaatiollinen tuki.

Tiimejä liiallisilta keskeytyksiltä ja kokouksilta suojelevan tiimin kontaktihenkilön

koettiin edistävän tiimien kykyä itse hallinnoida työnsä suoritusta ja aikataulua. Tiimin kontaktihenkilö suojeli tiimiä myös keskeytyksiltä edistäen syvää keskittymistä vaativaa ongelmanratkaisua koodatessa. (Tendedez et al. 2018; Moe, Dahl et al. 2019.) Tämä edisti tiimien valtaa Hackmanin funktioiden alueilla (3) ja (4) sillä se antoi tiimeille enemmän valtaa itsenäisesti hallinnoida työprosessiaan ja lähestyä työn tekemistä ilman häiriötekijöitä.

Tiimin koon rajaaminen hallittavaksi 7–9 hengen yksiköksi koettiin myös itsenäisen tiimin muodostumista edistäväksi käytännöksi (Mohagheghi et al. 2020). Tämän käytännön voidaan ajatella linkittyvän Hackmanin funktioihin (3) ja (4), koska pienemmällä tiimillä on lisääntyneesti valtaa itsenäisesti hallinnoida ja tehdä työtä ilman liian suuren tiimin sisäisiä riippuvuuksia ja koordinoitintarpeita.

Suunnanäyttäjinä toimineet kokeneet työyhteisön jäsenet auttoivat tiimejä hallinnoimaan itsenäisemmin työnsä suuntaa joutumatta nojaamaan liialti tiimien ulkopuolisiin auktoriteetteihin. (Moe, Dahl et al. 2019; Salameh ja Bass 2020). Tämä edisti tiimien valtaa Hackmanin funktiossa (1), sillä tiimit pystyivät suunnannäyttäjien avulla itsenäisemmin asettamaan työlleen suunnan.

Erilaiset tekniset ratkaisut olivat tärkeä osa tiimien autonomian edistämistä: yhdessä tapauksista itsepalvelualusta (eng. self-service platform) auttoi tiimejä itsenäisessä työskentelyssä, koska se mahdollisti tiimeille omien testiympäristöjen ja kokeilujen luomisen, ja koodin siirtämisen tuotantoon (eng. production) itsenäisesti (Gundelsby 2018). Toisessa tapauksessa automatisoidut testiympäristöt auttoivat tiimejä olemaan enemmän itse vastuussa ohjelmiston laadusta ja sen varmistuksesta (Moe, Šmite et al. 2021). Tekniset ratkaisut antoivat tiimeille lisääntyntä valtaa itse hallinnoida ja monitoroida työprosessiaan sekä toteuttaa työtä itsenäisesti linkittäen tämän menestystekijän Hackmanin funktioihin (3) ja (4).

## 5 Pohdinta

Ensimmäiseksi tässä luvussa pohditaan tulosten perusteella tasapainoa tiimien autonomian ja yhteisen linjauksen välillä vastaten tutkimuksen kolmanteen tutkimus-

kysymykseen *miten tiimien autonomiaa voidaan tasapainottaa tiimien ohjaamisen ja keskinäisen linjauksen kanssa?*. Toiseksi luvussa pohditaan mitä tulokset tarkoittavat käytännön kannalta, miten tulokset vertautuvat aiempiin tutkimuksiin ja mitkä olisivat aiheellisia jatkotutkimuskohteita. Kolmanneksi muotoillaan tulosten perusteella konkreettisia ohjeita johtajille edistämään tiimien autonomiaa ja sen tasapainotusta laajan skaalan ketterässä ohjelmistokehityksessä.

## 5.1 Tiimien autonomian tasapaino

Tasapaino tiimien autonomian, ohjauksen ja keskinäisen linjauksen välillä voisi löytyä yhteisömallisesta päätöksenteosta. Se samaan aikaan linjasi tiimejä keskinäisesti yhteensopiviin työn tekemisen käytäntöihin, edisti tiimien välistä informaation vaihtoa, mutta silti mahdollisti tiimien välisenä foorumina kyvyn vaikuttaa vahvasti yhteisön tiimejä koskeviin päätöksiin. Tätä johtopäätöstä tukee aineiston tutkimuksista T6-T8 ja T10. Tähän johtopäätökseen autonomian tasapainopisteestä tulevat myös Moe, Šmite et al. (2021).

Toinen tapa katsoa optimaalista tasapainon pistettä on suhteessa tiimien taitotasoon ja kokemukseen. Tulokset Petit ja Marnewickin (2019) tutkimuksesta viittaavat siihen, että hyvä käytäntö jakaa tiimeille lisääntynyttä autonomiaa on, kun tiimit edellisten työsuoritusten perusteella pystyvät hyödyntämään autonomiaa vastuullisesti ja tuottavasti. Tiimien autonomian tason nosto voi myös käänteisesti johtaa lisääntyneeseen kykyyn tehdä informoituja ja vastuullisia päätöksiä, jos tiimit samalla saavat tukea päätöksiin esimerkiksi valmennuksen ja tiimien välisen päätöksentekofoorumin muodossa (Salameh ja Bass 2020). Näin autonomian tasapainon voidaan ajatella nousevan kohti lisääntynyttä autonomiaa sitä mukaa, kun tiimit kartuttavat kokemusta ja työhön tarvittavia taitoja. Lisääntyneen autonomian antaminen tiimeille voi kiihdyttää niiden taito- ja kokemustason nousua, jos tiimit saavat samalla tarpeeksi tukea uusilla vastuualueillaan. Korkean itsehallinnon tason evääminen taitavilta ja kokeneilta tiimeiltä siis veisi pistettä liaksi kohti tiukkaa linjausta ja ylhäältä tulevaa kontrollia, kun taas itsehallinnon antaminen vähemmän taitaville tiimeille ilman tukea ja lisäresursseja veisi pistettä liaksi autonomian puolelle jolloin sen riskit liittyen esimerkiksi koordinaatioon, laadun varmistukseen

ja yhteisten tavoitteiden saavuttamiseen kasvaisivat.

## 5.2 Tulosten merkitys ja aiheet jatkotutkimukselle

Löydettyjen menestystekijäteemojen vähyys suhteessa löydettyihin haasteteemoihin voi implikoida, että haasteet autonomiasta ovat yleispätevämpiä laajan skaalan ketterässä ohjelmistokehityksessä kuin autonomiaa edistävät menestystekijät. Menestystekijät voivat olla enemmän riippuvaisia organisaation kontekstista eli esimerkiksi siitä minkälaista ohjelmistoa organisaatio kehittää, minkälainen kulttuuri organisaatiossa vallitsee ja mikä on organisaation ketterien tiimien nykyinen taitotaso. Tätä havaintoa tukee se, että yksi menestystekijäteemoista oli autonomiaa edistävät käytännöt, jonka sisältämät menestystekijät olivat melko yksilöllisiä tutkimuksissa seuratuissa tiimeissa ja organisaatioissa. Yhteisöllistä suunnittelua ja päätöksentekoa voidaan pitää yhtenä autonomiaa edistävänä käytäntönä, joka nousi esiin aineiston tutkimuksista. Myös sen onnistunut toteuttaminen todennäköisesti riippuu organisaation kontekstista ja siitä mitä haasteita organisaatio haluaa sillä ratkaista edistääkseen ja tasapainottaakseen tiimien autonomiaa.

Neljä kuudesta haasteteemasta linkittyivät Hackmanin funktioon (3), joka voidaan mieltää perinteisesti keskijohdon alueeksi; Hackman (1986) itse nimittää työyksiköitä, joilla on valtaa työn funktioista vain työn suorittamisen alueella *johtajavetoisiksi työyksiköiksi* (eng. manager-led unit). Nämä tulokset ovat linjassa Dikertin et al. (2016) löydösten kanssa, jossa yksi haastetekijöistä ketteriin menetelmiin siirtyessä laajassa skaalassa oli keskijohdon vaikeus päästää irti käskevästä ja hallitsevasta johtamistyylistä. Samoin kaikki menestystekijäteemat edistivät tiimien auktoriteettia juuri funktion (3) alueella.

Yllättävänä löytönä voidaan pitää sitä, kuinka moni haastetekijä vei valtaa tiimeiltä itse työn teosta Hackmanin (1986) funktiossa (4). Haasteissa oli tavallista, että ketterien tiimien piti käyttää ylhäältä tulleita käytäntöjä ja prosesseja vieden heiltä valtaa lähestyä itse työn suorittamista itsenäisesti. Tämä on luultavasti helpoin alue, jossa organisaatiot voivat edistää ketterien tiimien autonomiaa keräten autonomiasta koituvia hyötyjä ilman, että autonomian tason nosto tällä työn

funktion alueella aiheuttaisi merkittäviä ongelmia tiimien keskinäiseen linjaukseen. Tälle havainnolle oletuksena on, että suunta tehtävälle työlle on määritelty selkeästi.

Jotta tulokset olisivat paremmin yleistettävissä, tulisi tarkasteltavien tutkimusten määrää lisätä. Autonomiata tulisi tarkastella myös niiden laajan skaalan ketterän ohjelmistokehityksen tutkimusten pohjalta, jotka eivät erityisesti keskity tarkastelemaan aiheita tiimien autonomian perspektiivistä. Hackmanin (1986) funktiot antavat myös vain yhden näkökulman tiimien autonomiaan. Kokonaisvaltaisen ymmärryksen luomiseksi haaste- ja menestystekijöitä ja autonomian tasapainoa tulisi tarkastella useamman eri teorian näkökulmasta.

Tutkimuksen tulosten perusteella jatkotutkimusta tulisi tehdä erityisesti yhteisömallisesta päätöksenteosta autonomian edistäjänä ja mahdollisena tasapainottajana tiimien keskinäisen linjauksen välillä. Tutkittavaa olisi esimerkiksi siinä mitkä perinteisen johdon tehtävät yhteisömallinen päätöksenteko voisi korvata, ja kykenisikö se toteuttamaan ne paremmin tai samalla tasolla edistäen samaan aikaan tiimien autonomiaa. Toinen jatkotutkimusaihe voisi olla autonomiaa edistävien menestystekijöiden tarkastelu erityisesti organisaatioiden ja tiimien kontekstin näkökulmasta. Eli miten esimerkiksi tekijät kuten minkälaista ohjelmistoa organisaatio kehittää, millä alalla organisaatio toimii ja minkälainen organisaation kulttuuri on vaikuttavat siihen miten organisaation tulisi käytännössä lähestyä autonomian edistämistä tiimeissä

### 5.3 Käytännön ohjeita johtajille

Tutkimuksen tulosten pohjalta voidaan muotoilla seuraavia käytännön ohjeita laajan skaalan ketterän ohjelmistokehitystyön johtamiseen:

- 1) Varmista, että tiimeillä on autonomian vähimmäisvaatimukset kunnossa. Eli tiimien sisältä täytyy löytyä itsenäiseen työhön tarvittava osaaminen, ja ympäröivän organisaation pitää tukea tiimejä mahdollistamalla itsenäiseen työhön tarvittavat resurssit.
- 2) Pyri luomaan autonomiaa kunnioittava työympäristö minimoimalla tiimeihin

kohdistuvat ulkoiset häiriötekijät kuten sprintin kehitysjonon ulkopuolelta tulevat akuutit tehtävät.

3) Varmista selvyys tiimien vastuualueista ja rooleista. Osallista tiimejä tavoitteiden määrittelyyn tai kommunikoi vastuualueet ja roolit erityisen huolellisesti varmistaen niiden selvyys suoraan tiimeiltä.

4) Fasilitoi tiimien yhteisöllistä ja itsenäistä päätöksentekoa. Anna etenkin päätökset työn tekemisen käytännöistä ja työn hallinnoimisesta tiimien keskinäisten foorumien vastuulle. Luota tiimeihin ja anna heille valtuuksia tehdä itsenäisiä päätöksiä varsinkin työn toteuttamisen alueella.

5) Rajaa tiimeille mahdollisimman itsenäiset työalueet. Tunnista ja kitke aktiivisesti riippuvuuksia muihin tiimeihin ja ympäröivään organisaatioon.

6) Tunnista autonomiaa edistävien käytäntöjen kontekstuaaliset rajoitteet. Muiden organisaatioiden autonomiaa edistävät käytännöt eivät välttämättä toimi juuri sinun organisaatiossasi. Näin on tarpeellista olla avoin uusille nouseville käytännöille ja etsiä aktiivisesti yhteistyössä tiimien kanssa oman organisaation kontekstiin sopivia käytäntöjä autonomian edistämiseksi ja tasapainottamiseksi.

## 6 Yhteenveto

Ensimmäiseksi tämän tutkimuksen tarkoituksena oli selvittää haaste- ja menestystekijöitä tiimien autonomialle laajan skaalan ketterässä ohjelmistokehityksessä. Toiseksi tavoitteena oli selvittää miten autonomia voidaan tasapainottaa tiimien ohjaamisen ja keskinäisen linjauksen välillä, jotka ovat välttämättömiä laajan skaalan kontekstissa. Löydetyt haasteteemat olivat tarvittavan osaamisen ja resurssien puute, ulkoa tulevat häiriötekijät, ulkoa tulevat linjauksen työhön ja tiimin rakenteeseen, vaikeus rajata tiimien työtä itsenäisiin osiin, vähäinen luottamus ja valtuutukset päätöksiin ja epäselvyys yhteisistä tavoitteista ja rooleista. Löydetyt menestystekijäteemat olivat yhteisöllinen suunnittelu ja päätöksenteko, työn rajaaminen itsenäisiksi kokonaisuuksiksi, luottamus tiimeihin ja valtuudet tehdä päätöksiä ja autonomiaa edistävät käytännöt. Haasteet esiintyivät laajalti Hack-

manin (1986) organisaatiollisen työn funktiossa (3), mikä osaltaan vahvistaa aiempia löytöjä, jossa johdon on vaikea sopeutua uuteen ketterään ympäristöön organisaatiossa ja antaa ketterille tiimeille tilaa tehdä itsenäisesti työtä. Tasapaino autonomian ja keskinäisen linjauksen välillä voisi löytyä yhteisöllisestä päätöksenteosta tiimien välillä. Tasapainopisteen autonomian ja keskinäisen linjauksen välillä voidaan ajatella nousevan kohti lisääntynyttä autonomiaa tiimien osaamisen ja kokemuksen karttuessa. Osaamisen kehittymistä voidaan myös käänteisesti edistää antamalla tiimeille lisää autonomiaa tukien heitä samalla esimerkiksi tiimejä valmentamalla ja luomalla yhteisöllisen päätöksenteon foorumi tiimien päätösten tueksi. Aihetta jatkotutkimukselle olisi erityisesti yhteisömallisesta päätöksenteosta autonomian tasapainottajana ja organisaatiollisten kontekstien vaikutuksesta autonomian menestystekijöihin.

## Viitteet

- Ambler, Scott W ja Mark Lines (2012). *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM press.
- Beck, Kent, Mike Beedle, Arie Van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries et al. (2001). *The agile manifesto*. URL: <https://agilemanifesto.org/>.
- Boehm, Barry ja Richard Turner (2005). "Management challenges to implementing agile processes in traditional development organizations". *IEEE software* 22.5, s. 30–39.
- Deci, Edward L. ja Richard M. Ryan (2012). "Self-determination theory." Teoksessa: *Handbook of theories of social psychology*, Vol. 1. Thousand Oaks, CA: Sage Publications Ltd, s. 416–436. ISBN: 978-0-85702-960-7 (Hardcover). DOI: [10.4135/9781446249215.n21](https://doi.org/10.4135/9781446249215.n21). URL: <https://doi.org/10.4135/9781446249215.n21>.
- digital.ai (2021). *15th State of Agile Report*. URL: <https://digital.ai/resource-center/analyst-reports/state-of-agile-report>.
- Dikert, Kim, Maria Paasivaara ja Casper Lassenius (2016). "Challenges and success factors for large-scale agile transformations: A systematic literature review". *Journal of Systems and Software* 119, s. 87–108.
- Dingsøy, Torgeir, Nils Brede Moe, Tor Erlend Fægri ja Eva Amdahl Seim (2018). "Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation". *Empirical Software Engineering* 23.1, s. 490–520.
- Dybå, Tore ja Torgeir Dingsøy (2008). "Empirical studies of agile software development: A systematic review". *Information and software technology* 50.9-10, s. 833–859.
- Fenton-O'Creevy, Mark (1998). "Employee involvement and the middle manager: evidence from a survey of organizations". *Journal of Organizational Behavior: The International Journal of Industrial, Occupational and Organizational Psychology and Behavior* 19.1, s. 67–84.



- Gundelsby, Jan Henrik (2018). "Enabling autonomous teams in large-scale agile through architectural principles". Teoksessa: *Proceedings of the 19th International Conference on Agile Software Development: Companion*, s. 1–4.
- Gustavsson, Tomas (2019). "Voices from the teams-impacts on autonomy in large-scale agile software development settings". Teoksessa: *International Conference on Agile Software Development*. Springer, s. 29–36.
- Guzzo, Richard A ja Marcus W Dickson (1996). "Teams in organizations: Recent research on performance and effectiveness". *Annual review of psychology* 47.1, s. 307–338.
- Hackman, J. Richard (1986). "The psychology of self-management in organizations." Teoksessa: *The Master lectures*, Vol. 5. Washington, DC, US: American Psychological Association, s. 89–136. ISBN: 0-912704-48-9 (Paperback). DOI: [10.1037/10055-003](https://doi.org/10.1037/10055-003). URL: <https://doi.org/10.1037/10055-003>.
- Hoegl, Martin ja Praveen Parboteeah (2006). "Autonomy and teamwork in innovative projects". *Human Resource Management: Published in Cooperation with the School of Business Administration, The University of Michigan and in alliance with the Society of Human Resources Management* 45.1, s. 67–79.
- Langfred, Claus W (2000). "The paradox of self-management: Individual and group autonomy in work groups". *Journal of Organizational Behavior* 21.5, s. 563–585.
- Larman, Craig (2010). *Practices for scaling lean & Agile development: large, multisite, and offshore product development with large-scale scrum*. Pearson Education India.
- Leffingwell, Dean (2007). *Scaling software agility: best practices for large enterprises*. Pearson Education.
- Livermore, Jeffrey A (2008). "Factors that Significantly Impact the Implementation of an Agile Software Development Methodology." *J. Softw.* 3.4, s. 31–36.
- Mathieu, John, M Travis Maynard, Tammy Rapp ja Lucy Gilson (2008). "Team effectiveness 1997-2007: A review of recent advancements and a glimpse into the future". *Journal of management* 34.3, s. 410–476.
- Mikalsen, Marius, Magne Næsje, Erik André Reime ja Anniken Solem (2019). "Agile Autonomous Teams in Complex Organizations." Teoksessa: *XP Workshops*, s. 55–63.

- Moe, Nils Brede, Bjørn Haug Dahl, Viktoria Stray, Lina Sund Karlsen ja Stine Schjødt-Osmo (2019). "Team autonomy in large-scale agile". Teoksessa: *Proceedings of the Annual Hawaii International Conference on System Sciences (HICSS)*. AIS Electronic Library, s. 6997–7006.
- Moe, Nils Brede, Darja Šmite, Maria Paasivaara ja Casper Lassenius (2021). "Finding the sweet spot for organizational control and team autonomy in large-scale agile software development". *Empirical Software Engineering* 26.5, s. 1–41.
- Mohagheghi, Parastoo Omang, Casper Lassenius ja Ingrid Omang Bakken (2020). "Enabling team autonomy in a large public organization". Teoksessa: *Agile Processes in Software Engineering and Extreme Programming–Workshops*, s. 245.
- Olsson Holmström, Helena ja Jan Bosch (2016). "No more bosses?: A multi-case study on the emerging use of non-hierarchical principles in large-scale software development". Teoksessa: *17th International Conference on Product-Focused Software Process Improvement, Trondheim, Norway (November 22nd-24th 2016)*. Springer, s. 86–101.
- Paasivaara, Maria (2017). "Adopting SAFe to scale agile in a globally distributed organization". Teoksessa: *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*. IEEE, s. 36–40.
- Petersen, Kai ja Claes Wohlin (2010). "The effect of moving from a plan-driven to an incremental software development approach with agile practices". *Empirical Software Engineering* 15.6, s. 654–693.
- Petit, Yvan ja Carl Marnewick (2019). "Earn your wings: a novel approach to deployment governance". Teoksessa: *International Conference on Agile Software Development*. Springer, s. 64–71.
- Šāblis, Aivars ja Darja Šmite (2016). "Agile teams in large-scale distributed context: Isolated or connected?" Teoksessa: *Proceedings of the Scientific Workshop Proceedings of XP2016*, s. 1–5.
- Salameh, Abdallah ja Julian M Bass (2020). "Spotify tailoring for architectural governance". Teoksessa: *International conference on agile software development*. Springer, s. 236–244.
- Sommerville, Ian (2016). *Software engineering*. eng. Tenth Global edition. Always learning. Boston: Pearson. ISBN: 9781292096148.

- Tendedez, Helena, Maria Angela Felicita Cristina Ferrario ja Jonathan Nicholas David Whittle (2018). "Software Development and CSCW: Standardization and Flexibility in Large-Scale Agile Development". *Proceedings of the ACM on Human-Computer Interaction-CSCW* 2.CSCW.
- Uludag, Ömer, Pascal Philipp, Abheeshta Putta, Maria Paasivaara, Casper Lassenius ja Florian Matthes (2020). "Revealing the state-of-the-art in large-scale agile development: A systematic mapping study". *arXiv preprint arXiv:2007.05578*.