

Riku Vidqvist

**SAFE-VIITEKEHYKSEN KÄYTTÖNOTTO:
MAHDOLLISUUDET JA HYÖDYT**



JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2021

TIIVISTELMÄ

Vidqvist, Riku

SAFe-viitekehysten käyttöönotto: mahdollisuudet ja hyödyt.

Jyväskylä: Jyväskylän yliopisto, 2021, 32 sivua.

Tietojärjestelmätiede, kandidaatintutkielma

Ohjaaja(t): Halttunen, Veikko

Ketterät menetelmät ovat saaneet alkunsa virallisesti vuonna 2001 kirjoitetusta Agile Manifestosta. SAFe-viitekehys on organisaatioon implementoitava skaalautuva ketterä kehitysmenetelmä, jonka pyrkimyksenä on mahdollistaa ketterien kehitysmenetelmien implementointi suurempiin organisaatioihin, perinteisen tiimikoon sijasta. Ketterien menetelmien skaalaaminen on haastavaa, koska menetelmät on alun perin tarkoitettu pienille, yhden kehitystiimin sisältäville organisaatioille, joten SAFe:n tyyllisiä viitekehyskäytäntöjä ei löydy useita. Tutkielmassa tutkittiin SAFe-viitekehysten implementointia ja mitä hyötyjä sekä mahdollisuuksia se tarjoaa yrityksille. Tutkielma toteutettiin kirjallisuuskatsauksena, jonka aineistona oli SAFe perustajan Scaled Agile kirjoittamia raportteja, tutkimuksia viitekehysten implementoinnista sekä kirjallisuuskatsauksia liittyen aiheeseen. Scaled Agilen raporteissa huomioitiin positiivinen sävy omien tutkimusten suhteen sekä vertaisarvioimattomuus. SAFe-viitekehys sisältää useassa kohtaa Scrum-kehitysmenetelmän mukaista toimintaa, joka nähtiin tärkeänä osana prosessia. Tutkimuskysymykseni on: mitä mahdollisuuksia ja hyötyä SAFe-viitekehysten implementoinnista on yritykselle? Kirjallisuuskatsauksessa vertailtiin ja yhdistettiin Scaled Agilen kirjoittamia raportteja sekä muita tutkimuksia. Tutkimuksesta selvisi, että SAFe-viitekehysten implementointi parantaa erityisesti organisaation sisäistä kommunikaatiota sekä yhteistyötä eri osastojen välillä suurissa organisaatioissa. Onnistuneella implementoinnilla saavutetaan myös taloudellista hyötyä alentuneiden kustannusten johdosta. Myös julkaistun ohjelmiston laatu sekä uusien versioiden julkaisu kasvoi vertailtavana olevissa yrityksissä. Viitekehysten menestystekijäksi SAFe-implementaatioissa koettiin olevan laajan organisaation ketteröittäminen ja pienempien organisaatioiden suosimien prosessien skaalaaminen laajalaisempaan käyttöön. Onnistuneessa implementoinnissa organisaation sisäiset prosessit tehostuivat edellä mainituin tavoin, ja saavutettiin tehokkuutta, laatua sekä arvon nousua.

Asiasanat: ketterä kehitys, SAFe, skaalautuvat ketterät menetelmät, Scrum, lean-ajattelutapa, kehitys

ABSTRACT

Vidqvist, Riku

Implementing SAFe: pros and opportunities.

Jyväskylä: University of Jyväskylä, 2021, 32 pp.

Information systems' science, Bachelors' thesis

Supervisor(s): Halttunen, Veikko

Agile methods were officially initiated in 2001 when the Agile Manifesto was written. SAFe framework is a scalable agile framework implemented to an enterprise. Its goal is to enable traditional agile methods, such as Scrum, to a larger unit, rather than one team. Scaling agile methods is difficult, because they were initially created and meant for small ventures containing one development team. I researched the SAFe frameworks implementation and what it brings to an enterprise in terms of pros and opportunities. The research was conducted as a literature review. The material used for this literature review was Scaled Agile websites case-studies, research of the implementation process and other literature reviews. This thesis acknowledges the fact that non-peer reviewed articles usually present positive findings which may skew the results. SAFe framework contains activities resembling the Scrum method, which was seen as an important part of the developing process in SAFe. My research question was: what opportunities and pros does SAFe frameworks implementation enable for the enterprise? In this literature review the main findings were that, when implementing the SAFe framework, communication between teams and internally in the company was improved, and cooperation between teams was also improved. Successful implementation also achieves economic advantages such as lowered costs of development. Also, the quality of the product was improved and the rate at which newer versions of a system are shipped was higher. The success factors when implementing a SAFe framework was making a large organization agile and scaling the processes of smaller organizations to a more comprehensive use in a larger organization. When a successful implementation was reached, the previously mentioned factors were better and higher, and the internal processes of the organization became more efficient. Thus, organizations achieved efficiency, quality and added value.

Keywords: agile development, SAFe, scalable agile methods, Scrum, lean-mindset, development

KUVAT

KUVA 1 Scaled Agile Big Picture	14
KUVA 2 SAFe Essential	15
KUVA 3 SAFe Large Solution.....	18
KUVA 4 SAFe Portfolio	19
KUVA 5 SAFe Full.....	20

KUVIOT

KUVIO 1 Skaalautuvan ketterän menetelmän käyttöönotto	12
---	----

SISÄLLYS

TIIVISTELMÄ	2
ABSTRACT	3
KUVAT	4
SISÄLLYS	5
1 JOHDANTO	6
2 KETTERÄ OHJELMISTOKEHITYS	8
2.1 Ketterien menetelmien taustaa	8
2.2 Menetelmät	9
2.3 Skaalautuva Agile	10
3 SAFE-VIITEKEHYS	13
3.1 SAFe-viitekehyksen konfiguraatiot	15
3.1.1 Essential SAFe	15
3.1.2 Large-Solution SAFe	17
3.1.3 Portfolio SAFe	18
3.1.4 Full SAFe	20
3.2 Ketterä SAFe-organisaatio	21
3.3 Kehitystyö SAFe-viitekehyksen organisaatiossa	22
4 SAFE-VIITEKEHYKSEN MAHDOLLISUUDET JA HYÖDYT	24
4.1 Kommunikaatio	24
4.2 Tuottavuus ja produktiivisuus	25
4.3 Arvon tuottaminen	26
5 YHTEENVETO	28
LÄHTEET	30

1 JOHDANTO

Ketterät kehitysmenetelmät ovat luotuja perinteisesti yhdelle kehitystiimille ja tämän tiimin käytäntöjen parantamiseen. Monet ketterät menetelmät kuvataan pelkistetyn organisaation tueksi, eivätkä täten vastaa suoranaisesti todellisuuden tarpeita kuten laajempien projektin hallinnan haasteita. Ketterän lähestymistavan soveltamisessa skaalautuvasti on tyypillisesti käytetty "in-house" -metodeita, eli talon sisällä kehitettyjä projektinhallinnan menetelmiä eikä suoranaisia metodeita kuten Scrumia tai vastaavia. SAFe-viitekehys on luotu Lef-fingwell (2011) toimesta ratkaisemaan koon puolesta laajojen yritysten kehitystyön ongelmia. SAFe-viitekehys tarjoaa vaihtoehdon perinteisille tai yrityksen itse kehittämille kehitysmenetelmille. Se on ennalta määritelty viitekehys henkilöstön, prosessien ja organisaation ketteryyden suhteen. Pyrin tässä tutkielmassa selvittämään, mitä hyötyä kyseisen viitekehysten implementoinnista on yritykselle, ja mitä viitekehys mahdollistaa yritykselle. Tutkimus on rajattu hyötyihin ja mahdollisuuksiin. En kuitenkaan ota tutkielmassani kantaa SAFe-viitekehysten implementoinnin haasteisiin, mutta lukijan on syytä tiedostaa, että niitä on. Tutkielmaan sisältyviin kuviin on pyydetty käyttöoikeus Scaled Agilelta 18.05.2021 ja se myönnettiin Scaled Agilen johdosta 18.05.2021.

Haluan tutkia kyseistä aihetta, koska projektinhallinta on ollut ensisijainen kiinnostuksen kohteeni yliopisto-opintojen aikana, ja muita ketteriä menetelmiä on mielestäni tutkittu jo minua kiinnostavilta osin. On olennaista selvittää miten skaalautuvat ketterät menetelmät sekä erityisesti SAFe-viitekehys parantaa laajojen organisaatioiden toimintaa, ja millä tavoin. Tutkimusta aiheesta löytyy niukasti, mutta sen verran, että siitä saadaan koherentti kirjallisuuskatsaus. Suurin osa aikaisemmasta tutkimuksesta perustuu SAFe-viitekehysten implementoinnin menestystekijöihin (Ebert & Paasivaara, 2017; Paasivaara, 2017). Yliopiston tietojärjestelmätieteen kursseilla käydään tarkasti läpi mitä Scrum-metodologian mukainen kehitys on ja miten yksittäinen kehitystiimi toimii, mutta nämä kohdat eivät mielestäni sovellu laajan yrityksen toiminnan tarkasteluun ja optimointiin tehokkaimmalla mahdollisella tavalla. SAFe-viitekehysten tutkiminen on ollut erittäin mielenkiintoinen ja antoisa tehtävä, josta olen saanut suuren määrän uutta tietoa liittyen organisaationaaliseen ketteryyteen ja kyseisen kehitysmenetelmän positiivisiin puoliin. Jatkotutkimuk-

sen kannalta olisi mielenkiintoista päästä kirjoittamaan pro gradu -tutkielmaa yrityksestä, joka on implementoimassa SAFe-viitekehystä yritykseensä. Tällöin pääsisin aidosti seuraamaan ja tutkimaan implementointiprosessia. Skaalautuvien ketterien kehitysmenetelmien tutkiminen on olennaista, koska useilla yrityksillä on käytössään organisaation sisällä itse keksittyjä toimintatapoja monien eri tiimien välisen toiminnan organisoimiseen. Tutkimus hyödyttää yrityksiä, jotka eivät ole täysin varmoja mitä suoranaista hyötyä ulkoisen viitekehysten implementoinnista organisaation käyttöön on.

Tutkielmassani pyrin saamaan vastauksen seuraavaan tutkimuskysymykseen:

Mitä mahdollisuuksia sekä hyötyä SAFe-viitekehysten implementoinnista on yritykselle?

Tutkielma toteutettiin kirjallisuuskatsauksena. Tieteellisten artikkelien sekä julkaisujen lisäksi aineistona käytettiin Scaled Agile sivuston case-tutkimuksia viitekehysten implementoinneista. Tiedonhankinnassa hyödynsin erityisesti Google scholar tietokantaa sekä IEEE:ta. Tietoa haettiin tiivistelmän avainsanoissa olevin hakutermein sekä niiden erilaisten variaatioiden avulla. Tutkielman perusteella yhdistetyn tiedon avulla voidaan nimetä kolme osa-aluetta, joita SAFe-viitekehysten onnistunut implementointi hyödyttää tai tarjoaa lisää mahdollisuuksia: kommunikaatio, tuottavuus ja laatu sekä arvon tuottaminen. Viitekehyksellä on suuri merkitys organisaation kommunikaation parantumiseen. Kirjallisuuskatsausten ja tutkimusten perusteella miltei kaikki yritykset ovat nähneet parantumista kommunikaatiossa organisaation läpinäkyvyyden sekä osastojen välisen kommunikaation parantumisena. Myös tiimien välisen synkronaation, eli yhteistä päämäärää kohti tekeminen oli selkeämpää kuin aikaisemmin.

Tuottavuuden ja produktiivisuuden koettiin myös kasvaneen. Produktiivisuudella tarkoitetaan tässä asiayhteydessä laadukasta määrän kasvua, eli laadullisesti parempaa tuotetta nopeammin. Tämä oli merkittävä tekijä viimeisessä tarkasteltavassa osa-alueessa eli arvon nousemisessa. Jos yritys kykenee tuottamaan nopeammin laadukkaampaa ohjelmistoa, nousee yrityksen palveluiden arvo nopeammin. SAFe-viitekehysten voidaan siis sanoa tehostaneen tarkastelun alla olleiden yritysten sekä aikaisempien tutkimusten yritysten prosesseja monilla alueilla, tarjoten kokonaisvaltaista hyötyä toimintaan ja mahdollistaen esimerkiksi uusien järjestelmien tehokkaan implementoinnin. Yritykset tuottivat myös laadukkaampaa ohjelmistoa nopeammin, ja täten nostivat kannattavuuttaan ja tuloksellisuuttaan.

Tutkielma alkaa ketterien menetelmien käsitteiden määrittelyllä, mikä on olennainen osa tutkielmaa, jotta myöhempi teksti on ymmärrettävää. Tämän jälkeen erittellään viitekehysten ideologiaa ja sen osia. Kolmannessa sisältöluvussa tarkennetaan viitekehysten eri osien rakennetta ja roolien merkitystä, sekä miten itse kehitystyötä tehdään SAFe-viitekehysten mukaisessa organisaatiossa. Tämän jälkeen neljännessä sisältöluvussa pyrin vastaamaan tutkimuskysymykseeni kirjallisuuskatsauksen tavoin.

2 KETTERÄ OHJELMISTOKEHITYS

Ketterät kehitysmenetelmät ovat olleet ohjelmistojen kehittämistyöhön liittyvien projektien keskiössä Agile Manifeston julkaisusta vuodesta 2001 (Fowler & Highsmith, 2001. s. 28–35). Agile Manifesto tarjosi perinteiselle ja suositulle vesiputousmallille (Royce, 1978) vaihtoehdon sekä kiteytti muille metodeille, mitä ketteryydellä oikeasti tarkoitetaan. Myöhemmin suosiota kasvattanut Scrum-metodi on nykyään ohjelmistokehitystyön standardi (14th State of Agile -report, 2020). Ketterien menetelmien skaalautuvuudessa on haasteita (Carroll & Conboy, 2019, s. 44–50), mutta mitä mahdollisuuksia ja hyötyjä SAFe-viitekehyksen implementointi tuo yritykselle?

2.1 Ketterien menetelmien taustaa

Ketterä ohjelmistokehitys, Agile, on ohjelmajulistus käytänteistä, jotka auttavat tiimiä ajattelemaan ja työskentelemään tehokkaammin sekä tekemään parempia päätöksiä (Greene & Stellman, 2014). Aikaisemmin suosiossa olleet perinteiset ohjelmistokehityksen menetelmät keskittyvät etukäteen suunnitteluun ja tiukkaan muutoksen hallintaan, toisin kuin ketterät menetelmät, jotka suunniteltiin vastaanottamaan muutoksen ja tehokkaasti hallitsemaan sen (Highsmith & Cockburn, 2001. s. 120–127; Cockburn & Highsmith, 2001. s. 131–133). On myös esitetty todisteita siitä, miten ketterät menetelmät lisäävät asiakkaan sekä kehittäjien tyytyväisyyttä, mutta toisaalta myös siitä, miten ketterät menetelmät eivät välttämättä ole paras ratkaisu suuriin projekteihin (Dybå & Dingsøy, 2009. s. 6–9).

Boehmin tutkimuksessa (2002) muutokseen suhtautuminen oli menestystekijä sekä jossain tapauksissa osasy s epäonnistumiseen. Boehm (2002) ehdotti myös, että yrityksen tulisi löytää oma sisäinen tasapainonsa ketterille sekä perinteisille, suunnitelmallisuutta painottaville menetelmille. Virallinen kuvaus ketterästä ohjelmistokehityksestä sisältyy 2001 julkaistuun Agile Manifestoon, jonka kirjoittivat 17 ohjelmistokehityksen prosessien metodologista. Nämä 17 henkilöä edustivat parempaa tapaa kehittää järjestelmiä sekä ohjelmistoja ja

perustivat Agile Alliancen. ”Manifesto for Agile Software Development” on Agile Alliancen sivuilla luettavissa (Agile Alliance, 2021).

2.2 Menetelmät

Vuonna 2001 julkaistu Agile Manifesto (Fowler & Highsmith) painottaa ennen kaikkea iteroivaa prosessia ja ihmisten asettamista kehitystoiminnan tärkeimmäksi voimavaraksi. Nämä olennaiset ominaisuudet voidaan tiivistää Abrahamssonin, Salon, Ronkaisen sekä Warstan (2017) mukaan seuraavasti:

- Yksilöt ja kanssakäyminen prosessien ja työkalujen ylitse
- Toimiva ohjelmisto ennen kaiken kattavaa dokumentaatiota
- Yhteistyö asiakkaan kanssa yli sopimusneuvotteluiden
- Reagointi muutokseen suunnitelman mukaan kulkemisen sijasta

Listan vasemmalla puolella oleville asioille annetaan huomattavasti enemmän painoarvoa ketterässä kehityksessä kuin oikealla puolella oleville, vaikka ne ovatkin olennainen osa kehitystyötä (Chow & Cao, 2008. s. 961–971).

Agile Manifesto on luonnollisesti saanut monia piirteitä Scrumista, XP:stä sekä muista menetelmistä (Beck & Beedle ym., 2001), joiden kanssa henkilöt, agilistit, ketkä loivat Agile Manifeston, työskentelivät. On monia muita erilaisia ketteriä menetelmiä mitä voidaan luonnehtia ”ketteriksi”, mutta yleisesti kirjallisuudessa ketteriksi menetelmiksi mainitaan seuraavat: Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Crystal sekä Lean. Ketterille menetelmille on yleistä, että:

- Ne ovat tehtävien rakenteelta erilaisia (Nerur & Balijepally, 2007. s. 79–83; Thummadi ym. 2011. s. 67–76)
- Ne perustuvat uudelleen tehtäviin aktiviteetteihin (Pentland & Feldman, 2007. s. 781–795) kuten iteratiivisiin välimatkan maaleihin pyrkimiseen sekä päivän aloittavin tapaamisin (Schwaber & Beedle, 2001; Williams, 2012. s. 71–76).

Ketterät menetelmät pyrkivät siis ennen kaikkea iteroivaan prosessiin kehitystyössä. 14th Annual State of Agile -raportin (Digital.ai, 2019) mukaan 68 % vastanneista yrityksistä käyttää Scrumia eri tavoin. Scrum projektinhallintatyökaluna ohjelmistokehityksessä on erityisesti suosittu sen roolien ja selkeyden johdosta. Schwaber K. (2004) kuvailee Scrum-prosessin olevan erittäin virtaviivainen ja selkeä toiminnan, roolien ja päämäärän suhteen. Scrum ei kuitenkaan kuvaa mitä eri prosessin kohdissa tulee tehdä (Schwaber, 1997. s. 124), vaan se tarjoaa tekemiselle viitekehyksen ja mitä asioita roolilta odotetaan. Se sopii joustavuuden johdosta ketteriin ohjelmistokehitysprojekteihin, jossa esimerkiksi vaatimuksia voi tulla lisää sprintin sisällä (Schwaber, 1997. s. 124). Ohjelmisto-

kehitystyössä sprintillä tarkoitetaan tiettyä ajanjaksoa, jonka aikana tehdään sovittuja asioita ja jonka jälkeen seuraava versio ohjelmistosta julkaistaan (Agility.im, 2021). Scrum, XP ja Kanban ovat yhdessä miltei standardi ohjelmistokehitysalalla, ja niitä käytetään usein toisiaan täydentäen. Scrumin ja Kanbanin samanlaisuudet sallivat niiden käytön yhdessä ja toisiaan täydentäen, jolloin menetelmän termiksi muodostuu Scrumban. Se yhdistää iteratiivisen Scrum toiminnan sekä Kanbanin ominaisuudet kuten työmäärän visualisoinnin, tehtävänä olevan työmäärän rajoittamisen sekä "Lead Time"-mittaamisen (Nikitina, Kkajko-Mattson & Stråle, 2012. s. 140–149; Kniberg & Skarin, 2010). "Lead Time"-mittaamisella tarkoitetaan ominaisuuden tai tehtävän suorittamiseen kuluvaan aikaan kokonaisuudessaan eli ohjelmistokehityksen kontekstissa "backlogista" tuotantoon kuluvaan aikaan (Sugimori, Kusunoki, Cho, ym., 1977. s. 553–564). Näiden kahden menetelmän yhdistäminen mahdollistaa tiimin seuraamaan tarkasti ominaisuuksia, niiden suoritusaikaa sekä kehittämään iteratiivisesti uusia ominaisuuksia. Esimerkin mukaisesti Scrumban sopii erinomaisesti kehitystiimille, joka on ketterä ja omaa erinomaisen kommunikaation tiimin kesken sekä tuotteen omistajan kautta ulospäin muille sidosryhmille. Toimintatapa taas ei sovi monikansalliselle yhtiölle, joka omaa useita kehitysosastoja, joiden alla toimii taas useita kehitystiimejä. Kehityskäytänteissä on pakko ollut alkaa miettiä skaalautuvuutta, jos käytänteet tahdotaan laaja-alaisempaan käyttöön.

2.3 Skaalautuva Agile

Jotta voidaan tutkia tarkemmin mitä skaalautuva agile on, pitää ensin pystyä määrittelemään mitä "large-scale" agile on koon suhteen. Dingsøyr ym. (2014. s. 273–276) suorittamassa kirjallisuuskatsauksessa *koko* on aikaisemmin määritellyt:

- Henkilömääränä (työntekijät)
- Tiimien määränä
- Projektin budjetin mukaan
- Projektissa tuotetun lähdekoodin mukaan
- Projektin keston mukaan.

Dingsøyr ym. (2014) päätyivät seuraavaan määrittelyyn koon puolesta:

- Pienikokoinen: 1 tiimi
- Suurikokoinen: 2–9 tiimiä
- Erittäin suurikokoinen: 10+ tiimiä.

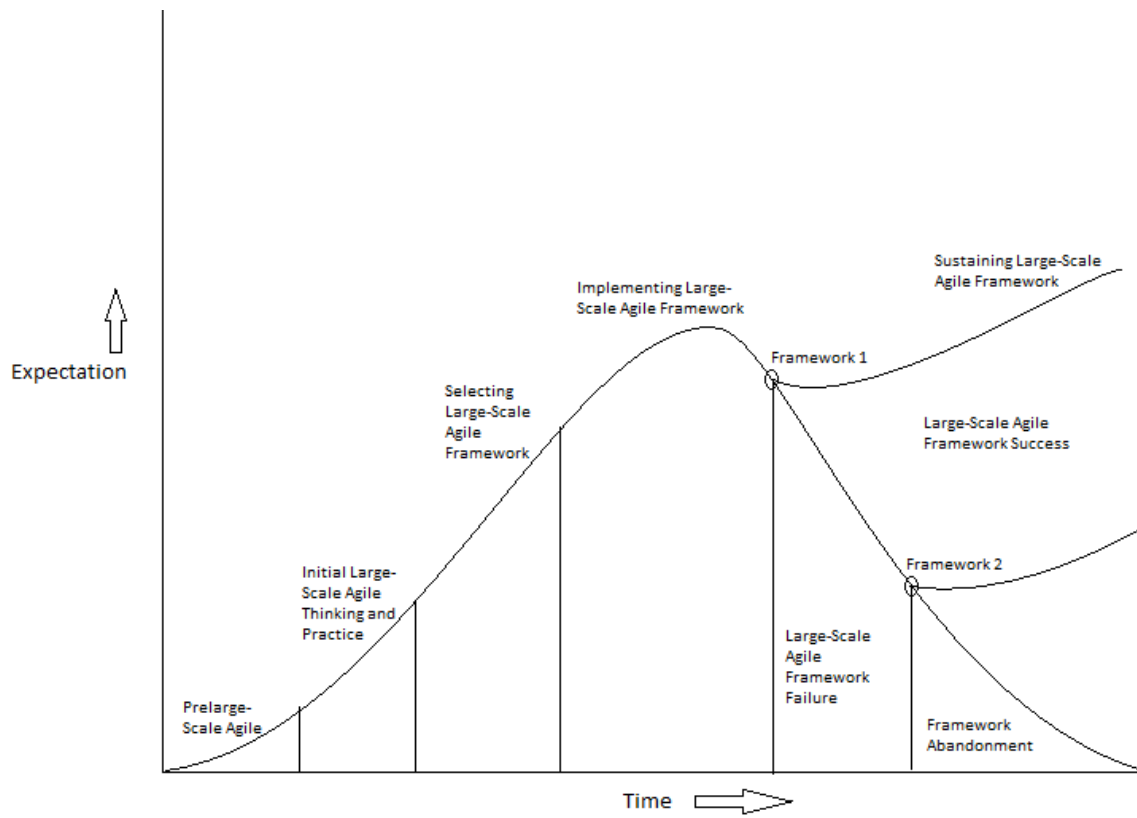
14th Annual State of Agile -raporttiin (Digital.ai, 2019) 35 % vastanneista käyttää skaalautuvana menetelmänään SAFe-viitekehystä. SAFe-viitekehityksen hyödyntäminen ohjelmistokehitystyössä on kasvanut vuosittain ja onkin nyky-

ään raportin mukaan suosituin skaalautuva Agile-viitekehys. Muita skaalautuvia, yleisesti käytettyjä menetelmiä ovat: Scrum of Scrums (SoS, 16 % aikaisempaan kyselyyn vastanneista), Large-Scale Scrum (LeSS, 4 % aikaisempaan kyselyihin vastanneista) sekä Disciplined Agile Delivery (DAD, 4 % aikaisempaan kyselyyn vastanneista) (Digital.ai, 2019).

Skaalautuva ketterä kehitys suurissa organisaatioissa on kompleksia ja sisältää suuren määrän erilaisia haasteita (Ebert & Paasivaara, 2017. s. 98-103). Suuret projektit vaativat oikeanlaista koordinaatiota sekä tiimien välistä kommunikointia. Lisäksi tiimien välisiä riippuvuuksia tulee hallinnoida, muut ei-ketterät yksiköt tulee ottaa huomioon (Saddington, 2012. s. 123-130) sekä oikeiden henkilöiden tulee olla osa prosessia (Moore & Spens 2008. s. 121-124; Conboy ym., 2011).

Haasteet johtuvat alun perin siitä, että ketterät menetelmät olivat suunniteltu käytettäväksi pienissä, yhden tiimin projekteissa (Boehm & Turner, 2005. s. 30-39). Dikertin, Paasivaaran ja Lasseniuksen tutkimus vuodelta 2016 kuvaillee erinäisiä syitä, miksi uuden, skaalautuvan ketterän menetelmän käyttöönotto on haastavaa: useimmiten syynä ovat vastahakoisuus uutta toimintatapaa vastaan tai työntekijöiden harjaannuttamisen vähyyys sekä näiden kanssa samankaltaiset ja yhdistyvät, organisaation toimintatapoihin vaikuttavat asiat. Yleisimmät menestystekijät onnistuneen ketterän skaalautumisen taustalla olivat henkilöstön kouluttaminen, tiedotus ja henkilöstön osallistaminen prosessiin sekä sellaisten tekijöiden sisällytys toimintaan, jotka puskevat prosessia eteenpäin (Chow & Cao, 2008; Ebert & Paasivaara, 2017).

Conboyn ja Carrollin (2019) julkaisivat esityksen tutkimuksessaan viitekehysten käyttöönoton haasteista sekä siitä miten useimmissa "Large-Scale Agile Framework" implementointiprosesseissa yleensä käy. Prosessi alkaa skaalaa-misvaihtoehtojen tutkimisella, eli erilaisten vaihtoehtojen etsinnällä sekä arvioinnilla. Tässä vaiheessa implementointia viitekehystä ei vielä tiedetä, mutta sille annetaan jo tiettyjä odottamia viitekehysten toiminnasta. Tästä vaiheesta siirrytään viitekehysten valitsemiseen, jolloin odotukset viitekehysten toimivuudesta kasvavat entisestään. Kohta, missä odotukset ovat korkeimmillaan on viitekehysten implementoinnin yhteydessä. Tämän jälkeen viitekehysten toiminta voi seurata viitekehys 1 tai viitekehys 2 mukaista toimintaa. Molemmat näistä implementointiyrityksistä voivat epäonnistua, mutta Conboyn ja Carrollin mukaan vasta toisen kerran yritys johtaa täydelliseen viitekehysten hylkäämiseen.



KUVIO 1 Skaalautuvan ketterän menetelmän käyttöönotto

3 SAFe-VIITEKEHYS

SAFe-viitekehys perustuu Lean-ajattelutapaan, jossa asiakas on keskeisimmässä osassa kaikessa tekemisessä ja arvoa tuotetaan vähentämällä ”jätettä” Lean-teorian mukaisesti (Poppedick, 2011. s. 1-7). Tarkastelen seuraavaksi SAFe-viitekehystä yleisimmiltä osilta, ja seuraavassa luvussa yksityiskohtaisemmin sen ominaisuuksia sekä vahvuuksia ohjelmistokehitystyössä.

SAFe-viitekehysten on perustanut vuonna 2011 Dean Leffingwell Scaled Agile -yrityksen yhteydessä. Yritys ylläpitää viitekehystä, sen dokumentaatiota sekä koulutusmateriaalia. SAFe-viitekehysten arkkitehtuuri korostaa kolmea organisaation perustasoa: tiimiä, ohjelmaa (eng. program) sekä portfoliota. Jokainen taso tekee omia tehtäviään ja on liitoksissa muihin (Leffingwell, 2018). Isommat yritykset voivat laajentaa edellä mainittua yhdellä lisätasolla: arvoketju. Jokaisen tason pohjana ovat tukevat elementit kuten SAFe-viitekehysten perusarvot sekä Lean-ajattelutapa. SAFe-viitekehysten määritelmä on erittäin tarkka ja joskus sisältää määritelmät jopa yksittäisille tapaamisille (Leffingwell, 2018).

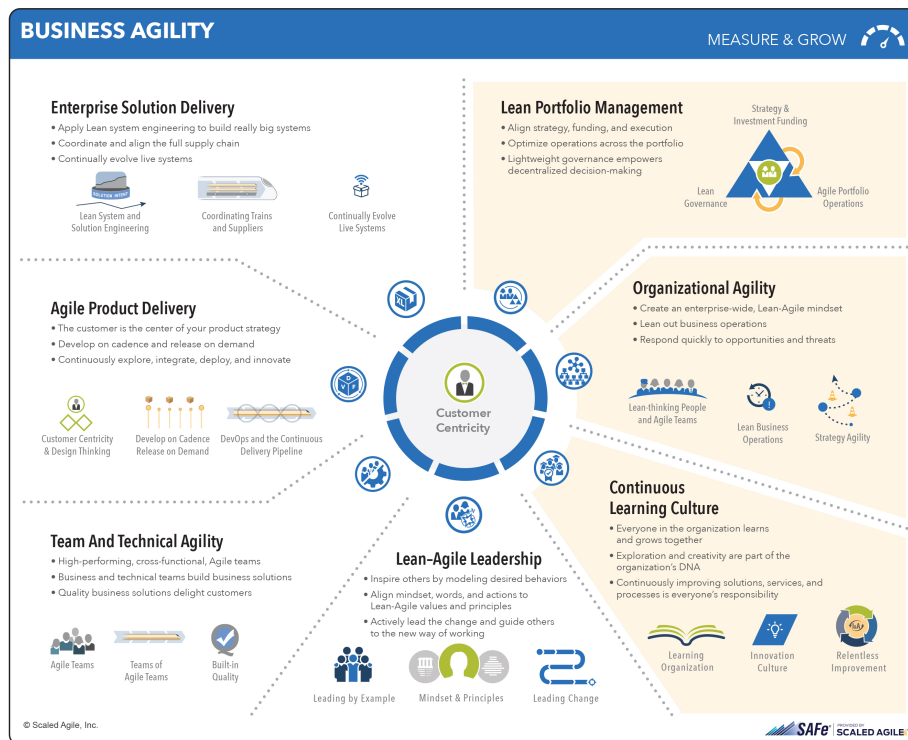
SAFe-viitekehyksessä olennaista on se, miten se vaikuttaa liiketoiminnan muihin edellä mainittuihin osa-alueisiin toisin kuin esimerkiksi perinteinen Scrum, joka keskittyy vain ohjelmistokehitykseen (Laanti, 2014). Perinteiset ketterät menetelmät eivät taivu kuvailemaan tehokkaimpia tapoja organisoida esimerkiksi portfolion ja ohjelman tasojen kesken (Laanti, 2014). SAFe-ottaa näissä tasoissa esimerkkiä Lean-ajattelumallista (Reinertsen, 2009) ja laajentaa viitekehystä yhteensopivammaksi tasojen kesken. Ennen kaikkea Lean tarjoaa SAFe-viitekehysten käyttöönoton yritykselle muuntautuvuutta ja tehokkuutta. Alla olevassa kuvassa (Kuva 2, SAFe) on yleiskatsaus SAFe-viitekehysten olennaisimmista osa-alueista. Keskellä on asiakas, joka ohjaa toimintaa. Scaled Agile sivuilla viitekehysten olennaisin asia, asiakaskeskeisyys, ”Customer Centricity”, määritellään monen eri tutkimuksen mukaisesti.

- Leonard ja Rayport (1997. s. 102-115) Harvard Business Review -artikkelissa esittävät käsitettä ”empaattinen design”, jolla tarkoitetaan omien ennakoideiden sivuuttamista ja ratkaisujen suunnittelua asiakkaan näkökulmasta.

- "Whole Product Thinking" -ajattelu (Theodore, 1980) on tärkeässä osassa luomassa perustaa SAFe-viitekehityksen asiakaskeskeisyydelle. Se kuvaillee millainen tuotteen tulisi olla ja mitä tasoja tuotteella on: yleinen, odotettu, muokattu ja potentiaalinen. Nämä tasot vaikuttavat asiakkaan odotusten ylittämiseen.

Muut olennaiset alueet kuvassa ovat:

- Lean-portfoliohallinta
- Organisaation laajuinen ketteryys
- Jatkuva oppimisen kulttuuri
- Lean-Agile johtajuus
- Tiimi ja tekninen ketteryys
- Ketterä tuotteen toimitus
- Yrityksen ratkaisun toimitus.



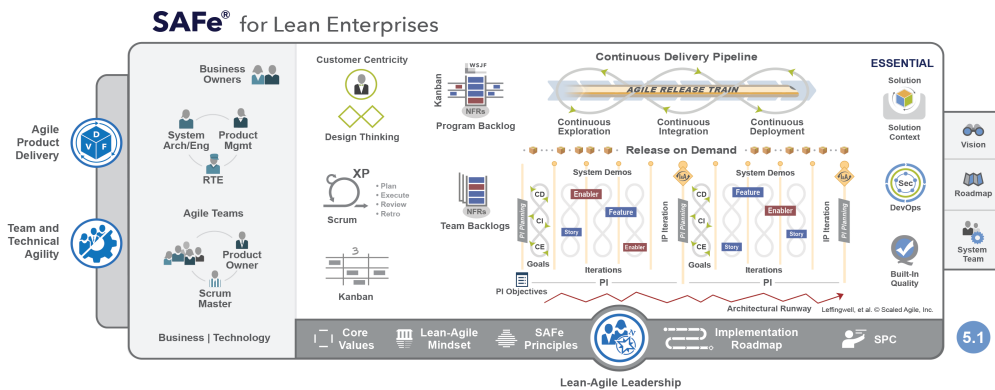
KUVA 1 Scaled Agile Big Picture

3.1 SAFe-viitekehyksen konfiguraatiot

Scaled Agile tarjoaa SAFe:a erilaisissa alkukonfiguraatioissa: "Essential", "Large Solution", "Portfolio" sekä "Full" SAFe. Konfiguraatiot soveltuvat eri kokoisille organisaatioille. Essential SAFe:n ollessa pohjaratkaisu, se sisältää kaikki olennaiset ja tärkeimmät aspektit SAFe-viitekehyksestä. Muut konfiguraatiot rakentavat Essential-mallin päälle lisäten tarpeen mukaan osa-alueita. Portfolio-malli ei esimerkiksi sovellu yhden suoranaisen ratkaisun tuottamiseen. Olenaisiin ominaisuuksiin SAFe-viitekehyksen mukaisessa organisaatiossa on ART-juna, eli Application Release Train. ART:t ovat äärimmäisen tärkeä osa jokaista SAFe-konfiguraatiota ja sen operatiivista toimintaa, riippumatta konfiguraation koosta. Ne ohjaavat kehitystyön tekemistä samaan suuntaan ja ajoittavat tiimien yhtäaikaista tekemistä oikein. ART-junilla pyritään siis suoranaisesti tiimien välisen työskentelyn synkronoimiseen.

3.1.1 Essential SAFe

Essential SAFe (kuva 2) on SAFe on yksinkertaisin aloituspiste SAFe-viitekehyksen implementointiin yritykseen. Essential SAFe sisältää kaikki olennaisimmat rakennusosat ja kuvailee kriittisimmät tekijät, jotta SAFe-viitekehys saadaan toimimaan halutulla tavalla ja tuoden kaikki viitekehyksen tarjoamat hyödyt (Leffingwell, 2018).



Essential-malli sisältää aiemmista mainituista kaksi olennaisinta, ohjelman (program) sekä tiimin. Yhdessä nämä tasot muodostavat peruskokonaisuuden organisaatorakenteelle, joka nimetään SAFe:n yhteydessä ”Agile Release Train” (lyh. ART). ART:ssa ketterät tiimit, tärkeät sidosryhmät sekä muut resurssit on määrätty käynnissä olevien tehtävien suorittamiseen (Leffingwell, 2018). Essential-malli tarjoaa olennaisimmat, ydinasiat SAFe-mallista. ART:t yhdistävät erilliset tekijät kuten sidosryhmät, johdon ja tiimit yhteiseen missioon yhtenäisen vision, tiekartan (roadmap) sekä tuotteen tehtäväkasauman (product backlog) mukaan. Tällöin tiimit pysyvät yhteisessä tavoitteessa parhaiten. Tiimien yhtenäiset kehitysiteraatiot eli sprintin aloittamisen sekä päätymisen ajankohdat ovat samat ja ne ovat saman kestoisia. Jokainen ART tuottaa täten laadukasta ja testattua tuotetta, kahden viikon välein. ”Program Increments (lyh. PIs)”, jotka tarjoavat pidemmän, kiinteän aikaikkunan suunnittelun, toteutuksen, tutkimisen sekä adaptoinnin inkrementeille. ART-junat koostuvat ja ylläpitävät jatkuvan kehityksen putkea (Continuous Delivery Pipeline), jonka avulla tuotetta kehitetään luoden pieniä arvon inkrementtejä. DevOpsin integraation ratkaisun tuottamiseen, joka mahdollistaa esimerkiksi kommunikointia, integraatioita sekä automaatioita.

Essential-malli sisältää myös viitekehykselle olennaisia sidosryhmiä ja henkilöstöä.

- **Järjestelmäarkkitehti/insinööri**
- **Tuotejohtaja**
- **Release Train Engineer (RTE)**
- **Liiketoiminnan Omistajat**
- **Asiakas**

Järjestelmäarkkitehti tai -insinööri on henkilö tai pieni tiimi, joka implementoi järjestelmäajattelua kaikkeen tekemiseen. Rooli määrittelee tuotteen arkkitehtuurin, määrittelee funktionaaliset sekä epäfunktionaaliset vaatimukset, päättää ratkaisun pääelementit sekä alijärjestelmät. Järjestelmä siis rakentuu tämän roolin määrittelyiden pohjalta. Tuotejohtaja toimii asiakkaan äänenä sisänpäin yrityksen henkilöstölle. Tuotejohtaja kommunikoi asiakkaiden tarpeita henkilöstölle, määrittelee järjestelmän ominaisuuksia ja osallistuu validointiin. Tuotejohtaja on myös vastuussa ohjelman tehtäväkasaumasta (program backlog) ja priorisoi ominaisuuksia ja mahdollisuuksia taloudellisella näkökulmalla. Rooli on siis hyvin samankaltainen Scrumin Tuotteen omistajan roolin kanssa. RTE:n tehtävänä on olla ”Scrum Master” yksittäiselle ART:lle. Rooli toimii kuten Scrum Masterin rooli, eli henkilö on vastuussa tuotteen tuottamisen onnistumisesta eli arvon tuottamisesta, mutta laajemmalla vastualueella. Rooliin sisältyy myös Scrum Masterin tavoin tiimin avustaminen, jotta se pääsee parhaaseen suoritukseensa. Liiketoiminnan omistajat ovat sidosryhmä, joka hoitaa pääsääntöisesti hallinnollisia sekä liiketoiminnallisia asioita liittyen ART:n tuottamaan ratkaisuun. Asiakas on luonnollisesti olennaisin osa arvon määrittelyä ja tuot-

tamista. Asiakkaalla on myös vastuita ratkaisun tuottamisessa, kuten vaatimusten määrittely ja palautteen antaminen.

Kolme olennaista aktiviteettiä, jotka auttavat koordinoimaan ART:ia

- **PI (Program Increments) suunnittelu**
- **Järjestelmädemo (System Demo)**
- **Muokkaantuminen ja tarkastelu**

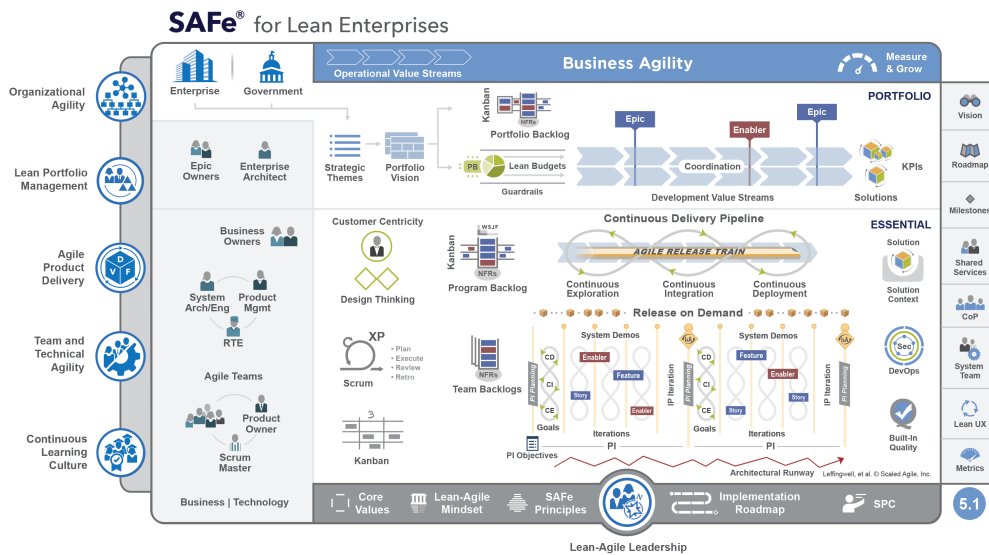
PI-suunnittelussa suunnitellaan seuraavan syklin toimintaa, jotta kaikki ART:t tekevät oikeita asioita yhteistä päämäärää kohden. Järjestelmädemo on esitysversio kyseisellä ajanhetkellä olevasta järjestelmästä. Iteraatioiden jälkeen demosta valmistuu aina seuraava versio, jonka avulla arvon tuottamista voidaan mitata. Tähän liittyy myös muokkaantuminen ja tarkastelu, joka on tapahtuma ART:lle, jolloin nykyistä ratkaisua demonstroidaan ja arvioidaan.

3.1.2 Large-Solution SAFe

Large-Solution SAFe tarjoaa aikaisempaan Essential-malliin lisää sisältöä. Large-Solution soveltuu tilanteisiin, jossa portfolionhallinta ei ole olennaista, vaan puhtaasti ratkaisun tuottaminen, kuten ilmailussa ja valtiollisissa projekteissa (Leffingwell, 2018). Alla olevassa kuvassa, jossa kuvataan Large-Solution SAFe:n toimintaa, esitetään prosessi Solution Trainin käyttämisestä, jota muissa ratkaisuihin ei ole. Konfiguraation ideana siis mahdollistaa tehoikkain mahdollinen tapa laajalle yritykselle tuottaa ratkaisuja.

nen, 2007. s. 56–65). Ketterät projekinhallintamenetelmät ovat lähtöisin ohjelmistokehityksen parista (Dybå & Dingsøyr, 2008. s. 833-859), joten SAFe:n yhteisen, organisaation kokoisen viitekehyksen käyttöönotto auttaa myös muiden liiketoiminnan aspektien ketteröittämisessä. Portfolionhallinnassa on olennaisinta (Martinsuo & Lehtonen, 2007):

1. Portfolion rahallisen arvon maksimointi
2. Yrityksen strategian yhdistäminen portfolioon
3. Projektien tasapainottelu portfolioon sisällä, resurssit huomioiden



SAFE® PROVIDED BY SCALED AGILE

KUVA 4 SAFe Portfolio

Portfoliomalli rakentaa Large-Solution -mallin tavoin Essential-mallin päälle, ja laajentaa mallia arvoketjuilla ja tarkemmilla budjeteilla sekä mittareilla. Olenaisimmat lisäykset ovat:

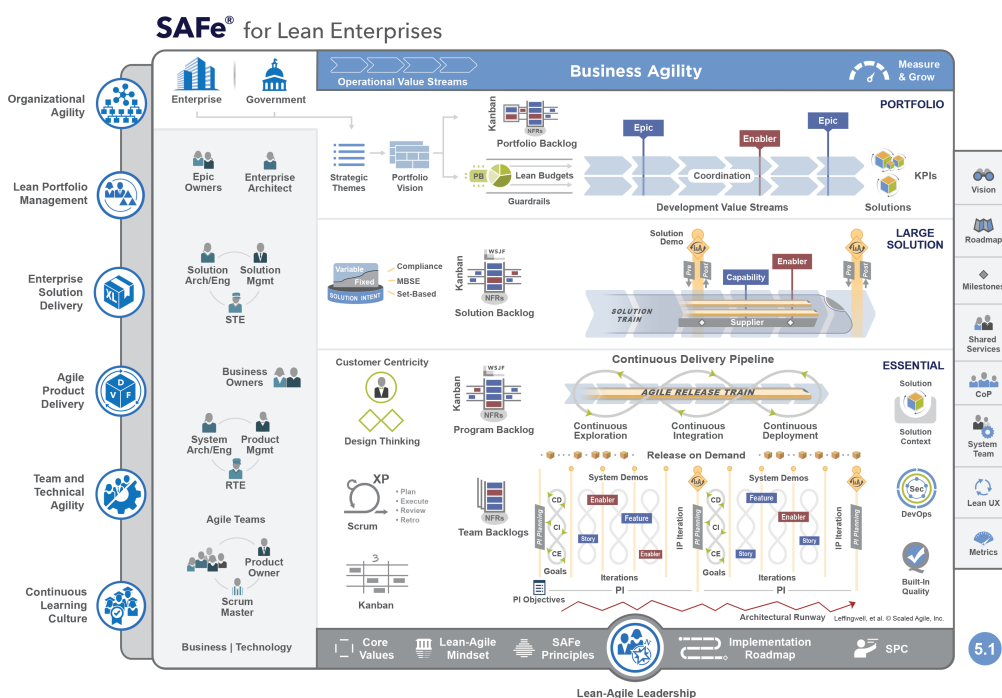
- **Arvoketjut**
- **Lean Portfolio Management**
- **Enterprise Architect**

Jokainen arvoketju omaa rahoituksen henkilöstölle ja ratkaisun tuottamiseen vaadituille resursseille, joilla luodaan arvoa asiakkaalle. Lean Portfolio Management henkilöstöön kuuluvat omaavat korkeimman tason päätösvallan SAFe-

portfolion sisällä. Enterprise Architectillä on kolme vastuualuetta: strategia ja investointivarojen hallinta, ketterän portfolion operaatiot sekä Lean-hallinta. Henkilö tai tiimi, jotka toimivat useiden arvoketjujen ja ohjelmien ylitse pyrkien mahdollistamaan strategista teknistä suuntaa, tavoitteena optimoida portfolion tuotokset.

3.1.4 Full SAFe

Full-malli sisältää kaikki yllä mainitut ominaisuudet sekä kaikki ensimmäisessä kappaleessa mainitut tasot. Myös alla oleva kuvio havainnoi miten laaja konfiguraatio on. Se sisältää kaikki ominaisuudet kuten ydinominaisuus ART:in, ratkaisun tuottamisen junan sekä portfolion hallinnan. Näiden yhteisellä tehokkaalla hyödyntämisellä yritys mahdollistaa koko SAFe:n tarjoamien ominaisuuksien hyödyntämisen. Malli on tarkoitettu yrityksille, jotka tuottavat ja ylläpitävät laajoja, integroituja ominaisuuksia, jotka vaativat satoja ihmisiä toimiakseen. Suurimmissa organisaatioissa useat instanssit erilaisilla SAFe-konfiguraatioilla ovat vaadittuja. Luonnollisesti myös suurin ja kattavin konfiguraatio tuo suurimman määrän henkilöstöä, joka on yksi tärkeimmistä osista viitekehysten implementoinnissa.



KUVA 5 SAFe Full

3.2 Ketterä SAFe-organisaatio

SAFe-viitekehys perustuu aiemmin määriteltyyn Lean-ajattelutapaan, jossa arvoa luodaan poistamalla ”jätettä” eli kaikkea sitä, mikä on epäolennaista. Lean-ajattelutapaan sisältyy viisi olennaista osaa: tuotantoflow, organisaatio, prosessienhallinta, metriikka sekä logistiikka (Feld, 2000. s. 4). Feld (2000) on avannut terminologiaa laajemmin seuraavasti:

- **Tuotantoflow** – Aspekti, joka koskettaa fyysisiä muutoksia sekä designstandardeja.
- **Organisaatio** – Aspekti, joka keskittyy tunnistamaan henkilöiden roolit ja funktiot organisaatiossa, uusien tapojen harjoittamisen sekä kommunikaation.
- **Prosessienhallinta** – Aspekti, joka monitoroi, hallinnoi, vakauttaa sekä etsii uusia tapoja tehostaa prosessia.
- **Metriikka** – Aspekti, joka seuraa näkyviä ja mitattavia tuloksia, kohdistettua parannusta sekä tiimin palkitsemista/tunnustuksen antamista.
- **Logistiikka** – Aspekti, joka antaa säännöt suunnittelulle ja materiaalien virralle.

SAFe-viitekehys siis pohjautuu ajattelutavaltaan 2000-luvun alun käytänteisiin palveluiden ja tavaroiden tuottamisesta. Edellisistä kohdista johdettuna kirja ”Lean Software Development” (Poppendieck & Poppendieck, 2003. s. 15-18) painottaa seitsemää lean-ohjelmistokehityksen periaatetta: optimoi kokonaisuus, tuhoa jäte, rakenna sisäistä laatua, opi jatkuvasti, toimita nopeasti, osallista kaikki ja jatka paremmaksi tulemista. Myös Abrahamsson, Ebert sekä Oza (2012. s. 22-25) ovat kuvailleet lean-ohjelmistokehityksen sisältävän seuraavat asiat, jotka kulkevat kehässä: luo arvoa asiakkaalle, tuhoa jätettä, optimoi arvoketjut, vahvista (empower) henkilöstöä sekä paranna jatkuvasti. Kaikissa kolmessa edellä mainitussa tavassa kuvailla lean-ohjelmistokehitystä yhteistä on henkilöstö, tulokset sekä toiminnan tehostaminen. Lean-ohjelmistokehityksessä pyritään siis henkilöstön voimaannuttamisen ja kouluttamisen avulla tuottamaan laadukkaita järjestelmiä, jotka tuottavat arvoa, täten oppien koko ajan mikä on tehokasta tuottamista, ja arvoa tuottavaa, ja mikä ”jätettä”. SAFe-viitekehysten implementointien organisaation täytyy mahdollistaa ja sitouttaa koko organisaation työskentely tämän ajattelumallin mukaan.

3.3 Kehitystyö SAFe-viitekehyksen organisaatiossa

SAFe-viitekehyksen mukainen organisaatio käyttää kehitystyössään hyödyksi ennen kaikkea lean-periaatteita, mutta myös ”Beyond Budgeting (Beyond Budgeting Principles, 2021)” sekä ”Blue Ocean (Kim, 2005. s. 105–121)”-periaatteita. Ensimmäinen näistä ohjaa johtajuutta ja prosesseja, toinen taas strategisia periaatteita. SAFe-viitekehyksen mukainen kehitystyö seuraa ART:en mukaista kehitystä. Tiimit asemoidaan yhteiseen teknologia- tai liiketoimintamission ja jokainen on teoreettisella tasolla oma organisaationsa (tyypillisesti 50–125 henkilöä), joka suunnittelee, sitoutuu, kehittää ja tuottaa yhdessä. ART:ien sisällä toiminta on iteratiivista, eli sisällä olevaa kehitysketjua toistetaan PI:n määräämän ajan, pyrkien toivottuun lopputulokseen. Ratkaisun, tuotteen sekä organisaation laajuudesta riippuen ART:ja voi olla useita työskentelemässä samaa kohdetta kohti. ART:n sisäinen iteratiivinen prosessi sisältää kuvailemisen, rakentamisen, validoinnin sekä julkaisemisen. Näin ART:ien prosessien kautta päästään jatkuvaan arvon virtaamiseen. (Scaled Agile, 2018).

ART:eilla on yhteiset periaatteet toimintaan:

- Aikataulu on tarkka ja siinä pysytään. Jos ominaisuus ei ehdi nykyiseen ”junaan”, se tuotetaan seuraavaan.
- Uusi järjestelmän kehitysinkrementti joka toinen viikko.
- Jokainen tiimi on synkronoitu yhteiseen PI-tavoitteeseen (Program Increment), ja niillä on yhteinen aloitus ja lopetusajankohta.
- Jokainen juna tietää miten paljon ominaisuuksia niiden pitää tuottaa yhdessä PI:ssä.
- Ketterät tiimit kehittävät ominaisuuksia ketterien menetelmien mukaisesti.
- Omistautuneet henkilöt
- Kasvotusten tapahtuva PI-suunnittelu
- Innovointi ja suunnittelu
- Inspektointi ja adaptiivisuus
- Kehitä rytmissä, julkaise pyydettyä.

Junat ovat organisoitu arvonluonnin ympärille. Perinteisessä funktionaalisessa organisaatiossa kehittäjät työskentelevät kehittäjien kanssa ja testajat testajien kanssa, joka luo organisaatiolle siloja osastojen välille. Siilot luovat kitkaa osastojen välille sekä vaikeuttavat yhteistyön tapahtumista ja onnistumista (Cilliers, Greyvenstein, 2012. s. 75–84). ART mahdollistaa järjestelmäajattelun ja arvon ympärille organisoimisen avulla perinteiset rajat ylittäviä tiimejä. Tämä luo ART:n sisälle enemmän lean-periaatteen mukaisia tiimejä, jolloin perinteistä projektinhallintaa ei tarvita, koska tiimeissä on huomattava määrä itseohjautu-

vuotta. ART:n sisäisillä tiimeillä on myös erilaisia vastuita ja tiimejä muodostetaan niiden toimintojen mukaan. ART:t koostuvat pääasiallisesti neljän erilaisen tiimin muodostamasta kokonaisuudesta.

1. "Stream-aligned team" - organisoitu työn virtaamiseen ja omaa kyvyn tuottaa arvoa suoraan asiakkaalle.
2. "Complicated subsystem team" - organisoitu erityisiä kykyjä vaativien alajärjestelmien käyttöä varten.
3. "Platform team" - organisoitu kehittämään järjestelmiä ja tukemaan muita tiimejä kehitystyön kanssa
4. "Enabling team" - organisoitu auttamaan muita tiimejä, omaavat usein erityisiä taitoja ja neuvovat uusien teknologioiden omaksumisessa.

ART:t vastaavat myös yleiseen ongelmaan monien tiimien ketterässä kehityksessä: tiimit, jotka työskentelevät saman ratkaisun eteen, toimivat itsenäisesti ja asynkronisesti. Tämän johdosta on erittäin haastavaa integroida järjestelmiä rutiininomaisesti. Tiimit siis integroivat, mutta järjestelmä ei. ART mahdollistaa tämän tapahtuvan synkronisesti ja tasaisin väliajoin sekä jokaisen iteraation lopussa tapahtuva järjestelmädemo auttaa myös tunnistamaan kehitystä ja mittaamaan sitä.

4 SAFe-VIITEKEHYKSEN MAHDOLLISUUDET JA HYÖDYT

Miten SAFe-viitekehyksen käyttöönotto vaikuttaa yrityksen toimintaan konkreettisesti? Esitän kolmea kategoriala hyötyjen ja mahdollisuuksien tarkastelulle: kommunikaatio, organisaatorakenne sekä taloudellinen hyöty. Tarkastelen viittä satunnaisesti valittua case-tutkimusta Scaled Agile -kotisivuilta, sekä aiheen aikaisemman tutkimuksen lähdekirjallisuutta. Yritykset, joiden case-tutkimukset valitsin, ovat:

1. Easterseals Northern California (Scaled Agile, 2020). Tekstissä ENC
2. Centers for Medicare & Medicaid services (Scaled Agile, 2019). Tekstissä CMM
3. Belastingdienst - Dutch Tax and Customs Administration (Scaled Agile, 2019). Tekstissä DTCA
4. Johnson Controls (Scaled Agile, 2018). Tekstissä JCI
5. EdgeVerve (Scaled Agile, 2018). Tekstissä EV.

Yritykset ovat eri toimialoilta, eli eivät vain järjestelmä- tai ohjelmistokehityksen yrityksiä. On huomioitava, että ei-vertaisarvioitujen artikkelit tyypillisesti esittävät positiivisia tuloksia (Zhou, Jin, Zhang ym., 2016, s. 153–160). On myös olennaista havainnoida, että onnistunut viitekehyksen implementointi paransi kaikkien case-tutkimusten yrityksillä sekä muissa tutkimuksissa kommunikaatiota.

4.1 Kommunikaatio

Osastojen välinen kommunikaatio nimetään suurien yritysten haasteeksi. ART-junien osastot muodostuvat erilaista osaamista omaavista henkilöistä, kuten kehittäjistä, testaajista ja suunnittelijoista. Ebertin ja Paasivaaran tutkimuksessa (2017) tutkittiin SAFe-viitekehyksen käyttöönoton yhteydessä yrityksen, Competel, kahta liiketoimintahaaraa. Ebert ja Paasivaara tunnistivat seitsemän menestys-

tekijää, jotka vaikuttavat onnistuneeseen implementointiin, ja joista voidaan vetää tiettyjä johtopäätöksiä viitekehyksen mahdollisuuksien suhteen. He havaitsivat, miten olennaista on kouluttaa, osallistaa sekä tiedottaa henkilöstöä implementaatiosta. Muutosagenttien käyttö havaittiin myös erittäin toimivaksi implementoinnin kannalta. Muutosagentilla tarkoitetaan henkilöä, joka aloittaa ja ohjaa muutosta organisaatiossa (Lunenburg, 2010, s. 1). Kokeneen ulkoisen konsultin käytöllä implementoinnin yhteydessä havaittiin olevan myös positiivinen vaikutus ja sen koettiin olevan yksi menestystekijä onnistumisessa. Ulkoinen SAFe-konsultti mahdollistaa viitekehyksen räätälöimisen sopivammaksi käyttöönottavalle yritykselle. Muut kohdat liittyivät myös olennaisesti kommunikointiin: dedikoitu RTE-henkilö, PI-suunnittelutapaamisten tarkkaa määrittely. Viimeisenä mainittiin tulosten tarkastelu ja analysointi. Tutkimuksessa yritys saavutti kaikki tavoitteensa:

- SAFe-viitekehys paransi läpinäkyvyyttä yrityksen sisällä ja paransi kommunikointia, vaikka yritys on kansainvälisesti operoiva.
- SAFe-viitekehys paransi synkronointia ja yhteistyötä tiimien välillä.
- SAFe-viitekehys paransi johtamista ja mahdollisti paremman, toimivamman organisaatorakenteen syntymisen.

Case-tutkimuksissa kommunikoinnin parantuminen havaittiin erittäin olennaisena osana organisaation toimintaa. Yrityksistä CMS raportoi henkilöstön tyytyväisyyden nousseen 27 % SAFe:n implementoinnin johdosta sekä DTCA esitti yleisellä tasolla parantuneen henkilöstön osallistumisen ja johtoportaan vähentymisen. Paasivaara (2017, s. 40) esitti SAFe-viitekehyksen implementaation parantaneen monikansallisen yrityksen kommunikointia PI-suunnittelutilaisuuksissa, vaikka tiimit sijaitsivat eri mantereilla (Malesia ja Suomi), eikä sijaintia pidetty haasteena. Ammarin ja Ghalian ym. (2017, s.126) laaja kirjallisuuskatsaus osoitti, että tyytyväisyys kommunikaatioon vaikutti positiivisesti organisaatioon sitoutumiseen. Pincus (1986, s. 395–419) tutki kommunikaation ja tyytyväisyyden yhteyttä, ja havaitsi vahvan korrelaation näiden kahden välillä. Edellisten tutkimusten perusteella voidaan johtaa seuraava johtopäätös: SAFe-viitekehyksen onnistunut implementointi parantaa henkilöstön tyytyväisyyttä työtä ja työyhteisöä kohtaan parantuneen kommunikoinnin avulla. Kommunikaation parantumisella on siis erittäin suuri merkitys välillisesti muuhunkin yrityksen toimintaan.

4.2 Tuottavuus ja produktiivisuus

Ebertin ja Paasivaaran tutkimus (2017) osoitti organisaatorakenteen merkityksen tehokkuuden ja tuottavuuden kannalta menestystekijäksi tutkimuksessaan. Moni case-tutkimuksen yrityksistä raportoi tehokkuuden tai tuotoksellisuuden noususta eri tavoin. ENC:lle SAFe-viitekehys mahdollisti it-osaston tuottavan parempaa laatua ennustettavammalla ja varmemmalla aikataululla. CMS:n laaturaportointiyksikkö esitti myös laadun kasvaneen: SAFe-viitekehyksen im-

plementoinnin seurauksena he saivat 55 % vähemmän yhteydenottoja aputilanteissa sairaaloilta, ja täten kasvattivat suoranaisesti asiakastyytyväisyyttä. DTCA raportoi laadun parantuneen ”ongelmien” vähentymisenä 80 %, ongelmalla tarkoitetaan vikaa järjestelmässä, sekä turvallisuusongelmien vähentyneen yhdessä divisioonassa 87 %. JCI raportoi julkaisevansa päivityksiä 2–4 kertaa useammin sekä järjestelmän virheiden määrä vähentyi kolme kertaa pienemmäksi, kun aiemmin. JCI:llä myös uusien ominaisuuksien tuottamiseen käytetty aika laski 50 %.

Brenner ja Wunder (2015 s. 1–2) havaitsivat myös samanlaisia tuloksia: SAFe:n ansiosta suunnittelu- ja kehitysprosessista tuli läpinäkyvämpää. Ebert ja Paasivaara (2017, s. 101–102) tunnustivat samanlaiset onnistumiset. Ebertin ja Paasivaaran mukaan (2017 s. 101–102) myös SAFe:n avulla saavutettiin useampia ja laadullisesti parempia julkaisuja odotettavammin. Tiimien muotoutuminen SAFe-viitekehityksen mukaisiksi nopeutti näiden päämäärien saavuttamisessa sekä selkeä tavoitteiden asettaminen jokaiselle ART-junalle ja PI:lle tehosti toimintaa yleisellä tasolla. SAFe:n avulla siis tuottavuutta nostettiin paremmalla suunnittelulla sekä oikeilla tiimien rakenteilla. Edellä mainitun kaltaiset parannukset ovat isoille organisaatioille merkittäviä kilpailuetuja esimerkiksi hintakilpailussa. Organisaatio pystyy tuottamaan ongelmaan ratkaisun nopeammin ja laadukkaammin kuin aikaisemmin, joten organisaatio kykenee myös myymään ratkaisuja samassa suhteessa kuin sen toiminta on tehostunut. Myös tuotettujen ratkaisujen laatu on erittäin olennainen kohta.

4.3 Arvon tuottaminen

SAFe-viitekehitys tarjoaa myös hyötyä myös taloudellisessa näkökulmassa. EV raportoi ”time-to-market” ajan laskeneen 12–18 kuukaudesta kuuteen kuukauteen suurien yritystason tuotteiden kanssa sekä pienempien tuotteiden kanssa kuudesta kuukaudesta kolmeen kuukauteen. Tämä tuo yritykselle merkittävää taloudellista hyötyä sen kyetessä tuottamaan arvoa 50 % nopeammin kuin ennen SAFe-implemентаatiota. EV raportoi myös kustannus per ominaisuus -mittarin laskeneen kahdeksan prosenttia per yksittäinen PI. Tämä on merkittävä taloudellisen hyödyn kasvu yrityksessä, jonka asiakkaita ovat 94 maassa toimivat pankit 848 miljoonan ihmisen asiakaskunnalla.

Arvon tuottamisen kehitys voidaan johtaa aikaisempien kohtien yhtäläisyyksistä. Ebert ja Paasivaara (2017) esittivät miten SAFe:n avulla yritys tuottaa parempaa ja laadukkaampaa tuotetta, ajallisesti nopeammin sekä parantuneella kommunikaatiolla yrityksen sisällä. Tämä tarkoittaa suoranaisesti sitä, että SAFe-implemентаatio mahdollistaa onnistuessaan yrityksen tuottaa arvoa tehokkaammin sekä nopeammin, lean-periaatteen mukaisesti. SAFe-pyrkii muuttamaan koko organisaation kulttuuria ketteräksi (Ebert & Paasivaara, 2017), jolloin edellä mainittu pääsee tapahtumaan.

Aikaisemmin esitettyjen SAFe-konfiguraatiokuvien perusteella voidaan myös todeta arvon tuottamisen tärkeys ja erityisesti sen tehostaminen SAFe-viitekehityksen mukaisessa organisaatiossa. Toiminta perustuu jokaisessa konfi-

guraatiossa laadukkaiden ratkaisuiden tuottamiseen mahdollisimman nopeasti rajatuilla resursseilla. Kappaleen alussa mainittu EV oli laskenut onnistuneen implementoinnin johdosta "time-to-market" aikaansa yli puolella. Jos suuren monikansallisen yrityksen liikevoitto-osuutta kasvattaa jokin muutos toiminnassa muutamalla prosenttiyksiköllä, on kyseessä valtava määrä varoja uusiin investointeihin. Ratkaisun tuottamiseen kuluvan ajan puolittaminen on merkittävä kilpailuetu ja arvon tuottaja kyseisen kokoluokan yrityksellä. Tämä sama päätelmä voidaan johtaa myös aikaisemmasta tuottavuuden tutkimisesta.

Kilpailuetua voi olla monenlaista, ja useimmiten kilpailu markkinoilla tapahtuu hinnan perusteella. Jos organisaatio voi SAFe-viitekehyksen implementoinnin johdosta kilpailla hinnalla sekä laadulla, on implementointi onnistunut. SAFe-viitekehys siis erityisesti hyödyntää sekä mahdollistaa erilaisten korrelaatioiden kautta organisaation tehokkuutta ja arvon tuottamisen kykyä.

5 YHTEENVETO

Tutkielmani aiheena oli SAFe-viitekehyksen käyttöönoton mahdollisuudet ja hyödyt. Tutkielmani pyrkimyksenä oli selvittää, mitä mahdollisuuksia SAFe tarjoaa viitekehyksen kautta yritykselle ja sen työskentelytavoille sekä mitä konkreettista hyötyä siitä on. Kirjallisuuskatsauksen tulokset esittävät seuraavia asioita mitä SAFe-viitekehyksen onnistuneella käyttöönotolla voidaan saavuttaa: parantunut kommunikaatio, työtyytyväisyys sekä tuotteen tuottamiseen vaikuttavat tekijät, kuten nopeus ja laatu. SAFe-viitekehys myös mahdollistaa yrityksille hyödyntää ketteriä menetelmiä koko organisaation laajuisesti. Tutkimieni artikkeleiden ja yrityksen case-tutkimusten yhdistämisen perusteella SAFe pyrkii yhtenäistämään sekä virtaviivaistamaan tiimien työskentelyä ja kasvattamaan tuotoksellisuutta parantaen yrityksen arvon tuottamisen kykyä tehokkaammin pienemmillä resursseilla. SAFe:n avulla suurimpaan hyötyyn organisaatiolle päästään kommunikaation suhteen (Brenner & Wunder, 2015; Paasivaara, 2017; Ebert & Paasivaara, 2017). Tutkimukseni perusteella hyödyille ja mahdollisuuksille voidaan esittää seuraavaa, loogista kulkua: parantuneesta kommunikaatiosta tiimien välillä päästiin työtyytyväisyyteen, jonka ansiosta tehokkuus ja tuottavuus nousivat. Näiden asioiden kasvu puolestaan vaikuttaa luonnollisesti arvon tuottamisen kykyyn, mikä yleensä mielletään laadun parantumisenä ja tuottamisen nopeutumisenä.

Toinen luku keskittyi avaamaan ketterien menetelmien historiaa ja sitä, mitä kaikkia erilaisia kehitystapoja on sekä mitä skaalautuvat ketterät menetelmät ovat. Ketterien menetelmien painotus on iteroivassa prosessissa ja ihmisten asettamisessa kehitystoiminnan tärkeimmäksi voimavaraksi. (Fowler & Highsmith, 2001). Skaalautuvien menetelmien ero perinteisiin menetelmiin on tiimien koossa ja muodostuksessa sekä erityisesti toimintatavoissa. Scrum kehitysmenetelmä on myös muokannut laajasti ketterää kehitystä ja sen implementointia yritykseen, ja on tarjonnut SAFe:lle laadukkaan pohjan viitekehyksen kehittämiseen.

Kolmas luku tarkasteli, millainen SAFe-viitekehyksen mukainen organisaatio on. SAFe-viitekehyksen mukaisia organisaatorakenteita on neljä erilaista riippuen yrityksen koosta. Luvussa esitettiin myös organisaatorakenteen osia ominaisuuksien sekä henkilöstön suhteen. Tärkeimpänä osana ovat ART-junat, jotka ohjaavat kehitystä yhdistäen tiimit ja selkeyttäen kommunikaatiota.

Neljäs luku esitti vastauksen tutkimuskysymykseen: mitä mahdollisuuksia ja hyötyjä SAFe-viitekehyksen käyttöönotto tarjoaa. Siinä esiteltiin, miten SAFe-viitekehyksen onnistunut implementointi tarjoaa parantunutta kommunikaatiota tiimien välillä, työtyytyväisyyttä, laadukkaampia ja nopeampia ominaisuuksien julkaisua sekä tehokkaampaa arvon tuottamista.

Etsiessä SAFe:n lähteitä ja siitä tarjolla olevaa tutkimusta, selvisi, miten paljon jatkotutkimusta aihe yleisellä tasolla vaatii. Scaled Agilen kotisivuilla julkaistut case-tutkimukset sisältävät luonnollisesti pelkästään positiivista faktaa ilman haittoja, mutta huomioin tämän tutkielmassani luvun neljä alussa listatessani tutkimukset mitä käytin. Skaalautuvia ketteriä menetelmiä on tutkittu erittäin niukasti ja mielestäni vertaisarvioituja case-tutkimuksia tarvittaisiin lisää aiheesta, joissa kerrottaisiin realistisesti haasteista mitä implementaatioprosessissa kohdattiin ja miten moni implementointia yrittäneistä yrityksistä epäonnistui. Lukijan on tärkeä ymmärtää, että tässä tutkimuksessa en ottanut miltei ollenkaan kantaa viitekehyksen implementoinnin haasteisiin tai kritiikkiin. Toki implementoinnin onnistuessa se mahdollistaa koko organisaation laajuisen ketterän toiminnan, joka on laajalle monikansalliselle yritykselle iso voimavara.

LÄHTEET

- Abrahamsson, P., Salo, O., Ronkainen, J. & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv Preprint arXiv:1709.08439*, Agility.im, *What is a sprint?* (25.3.2021) Haettu osoitteesta <https://agility.im/frequent-agile-question/what-is-a-sprint/>
- Ammari, G., Alkurdi, B., Alshurideh, A. & Alrowwad, A. (2017). Investigating the impact of communication satisfaction on organizational commitment: A practical approach to increase employees' loyalty. *International Journal of Marketing Studies*, 9(2), 113-133.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). Manifesto for agile software development.
- Beyond Budgeting (10.4.2021) Haettu osoitteesta <https://bbrt.org/what-is-beyond-budgeting/>
- Boehm, B. & Turner, R. (2005). Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, 22(5), 30-39.
- Brenner, R. & Wunder, S. (2015). Scaled agile framework: Presentation and real world example. (s. 1-2) IEEE. *2015 IEEE Eight International Conference. on Software Testing, Verification and Validation Workshops (ICSTW)*
- Chow, T. & Cao, D. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961-971.
- Cilliers, F. & Greyvenstein, H. (2012). The impact of silo mentality on team identity: An organisational case study. *SA Journal of Industrial Psychology*, 38(2), 75-84.
- Cockburn, A. & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133.
- Conboy, K., Coyle, S., Wang, X. & Pikkarainen, M. (2011). People over process: Key people challenges in agile development. *IEEE Software (PP 99)*, 1-1.
- Digital.ai, 14th Annual State of Agile Report (10.4.2021) Haettu osoitteesta <https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report>
- Dikert, K., Paasivaara, M. & Lassenius, C. (2016). Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*, 119, 87-108.

- Dingsøyr, T., Fægri, T. E. & Itkonen, J. (2014). What is large in large-scale? A taxonomy of scale for agile software development. (s. 273-276). *International Conference on Product-Focused Software Process Improvement*.
- Dyba, T. & Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5), 6-9.
- Dybå, T. & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859.
- Ebert, C., Abrahamsson, P. & Oza, N. (2012). Lean software development. *IEEE Computer Architecture Letters*, 29(05), 22-25.
- Ebert, C. & Paasivaara, M. (2017). Scaling agile. *IEEE Software*, 34(6), 98-103.
- Feld, W. M. (2000). *Lean manufacturing: Tools, techniques, and how to use them* CRC press.
- Fowler, M. & Highsmith, J. (2001). The agile manifesto. *Software Development*, 9(8), 28-35.
- Highsmith, J. & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, 34(9), 120-127.
- Kim, W. C. (2005). Blue ocean strategy: From theory to practice. *California Management Review*, 47(3), 105-121.
- Leffingwell, D. (2018). *SAFe 4.5 reference guide: Scaled agile framework for lean enterprises* Addison-Wesley Professional.
- Leonard, D. & Rayport, J. F. (1997). Spark innovation through empathic design. *Harvard Business Review*, 75, 102-115.
- Levitt, T. (1980). *Marketing success through differentiation-of anything* Graduate School of Business Administration, Harvard University. *Harvard Business Review* (January-February 1980).
- Lunenburg, F. C. (2010). Managing change: The role of the change agent. *International Journal of Management, Business, and Administration*, 13(1), 1-6.
- Martinsuo, M. & Lehtonen, P. (2007). Role of single-project management in achieving portfolio management efficiency. *International Journal of Project Management*, 25(1), 56-65.
- Moore, E. & Spens, J. (2008). Scaling agile: Finding your agile tribe. (s. 121-124) IEEE. *Agile 2008 Conference*.
- Nerur, S. & Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3), 79-83.

- Paasivaara, M. (2017). Adopting SAFe to scale agile in a globally distributed organization. (s. 36-40) IEEE. *12th International Conference on Global Software Engineering (ICGSE)*.
- Pentland, B. T. & Feldman, M. S. (2007). Narrative networks: Patterns of technology and organization. *Organization Science*, 18(5), 781-795.
- Pincus, J. D. (1986). Communication satisfaction, job satisfaction, and job performance. *Human Communication Research*, 12(3), 395-419.
- Poppendieck, M. (2011). Principles of lean thinking. *IT Management Select*, 18(2011), 1-7.
- Poppendieck, M. & Poppendieck, T. (2003). *Lean software development: An agile toolkit* Addison-Wesley.
- Saddington, P. (2012). Scaling agile product ownership through team alignment and optimization: A story of epic proportions. (s. 123-130) IEEE. *2012 Agile conference*.
- Scaled Agile, SAFe (28.3.2021) Haettu osoitteesta <https://www.scaledagileframework.com>
- Schwaber, K. (1997). Scrum development process. *Business object design and implementation* (s. 117-134) Springer.
- Schwaber, K. (2004). *Agile project management with scrum*. Microsoft press.
- Schwaber, K. & Beedle, M. (2002). *Agile software development with scrum*. Prentice Hall Upper Saddle River.
- Schwaber, K. & Sutherland, J. (2011). The scrum guide. *Scrum Alliance*, 21, 19.
- Stellman, A. & Greene, J. (2014). *Learning agile: Understanding scrum, XP, lean, and Kanban*. O'Reilly Media, Inc.
- Thummadi, B. V., Shiv, O. & Lyytinen, K. (2011). Enacted routines in agile and waterfall processes. (s. 67-76) IEEE. *2011 Agile Conference*.
- Williams, L. (2012). What agile teams think of agile principles. *Communications of the ACM*, 55(4), 71-76.
- Zhou, X., Jin, Y., Zhang, H., Li, S. & Huang, X. (2016). A map of threats to validity of systematic literature reviews in software engineering. (s. 153-160) IEEE. *23rd Asia-Pacific Software Engineering Conference (APSEC)*.